102 PROGRAMMI PER ATARI

600XL 800XL 130XE

DI JACQUES DECONCHAT

102 programmi per ATARI

600 XL, 800 XL, 130 XE

DELLO STESSO EDITORE

Volumi pubblicati

D. A. Lien - **Dizionario del Basic - Enciclopedia del linguaggio Basic**
J. Boisgontier - S. Brebion - **Il Basic per tutti - Iniziazione + programmi**
D.-J. David - **La scoperta del Commodore 64 - Vol. 1. Introduzione al Basic**
D.-J. David - **La pratica del Commodore 64 - Vol. 2. Periferiche e gestione dei file**
D.-J. David - **La pratica del Commodore 64 - Vol. 3. Linguaggio macchina e assembler del 6502**
J. Deconchat - **102 programmi per Commodore 64**
J. Boisgontier - **Commodore 64: metodi pratici**
J.-F. Séhan - **Supergiochi per il Commodore 64. 30 programmi in Basic.**
J. Boisgontier, S. Brebion, G. Foucault - **Il Commodore 64 per tutti**
D.J. David - **Chiavi per il Commodore 64**
V.C. Grandis, J.-F. Séhan - **Il Commodore 64 in famiglia**
X. Linant de Bellefonds - **La pratica dello ZX Spectrum - Vol. 1. Basic approfondito e introduzione al linguaggio macchina**
M. Henrot - **La pratica dello ZX Spectrum - Vol. 2. Programmazione in linguaggio macchina**
J.-F. Séhan - **Chiavi per lo ZX Spectrum**
J. Lévy - **Esercizi per lo ZX Spectrum**
J.-F. Séhan - **Alla ribalta: lo ZX Spectrum**
M. Henrot - J. Boisgontier - **Lo ZX Spectrum per tutti**
J. Deconchat - **102 programmi per ZX Spectrum e ZX 81**
C. Galais - **Vademecum per Applesoft**
J. Boisgontier - **L'Apple e i suoi files - Vol. 1. Metodi pratici**
B. De Merly - **Guida per l'Apple - Vol. 1. L'Apple standard**
B. De Merly - **Guida per l'Apple - Vol. 2. Le estensioni**
B. De Merly - **Guida per l'Apple - Vol. 3. Le applicazioni**
F. Lévy - **Esercizi per l'Apple II, II plus, IIe, IIc**
J. Boisgontier - **36 programmi per Apple II, IIe, II plus, IIc**
N. Bréaud-Pouliquen - **La pratica dell'Apple II - Vol. 1. Periferiche e gestione dei file**
N. Bréaud-Pouliquen - D.-J. David - **La pratica dell'Apple II, II plus, IIe - Vol. 2. Linguaggio macchina e assembler del 6502**
J. Boisgontier - **Apple IIe, IIc per tutti - Iniziazione più programmi**
J. Deconchat - **102 programmi per Apple II, II plus, IIe, IIc**
J. Rouault, P. Girard - **Pascal UCSD per Apple II - Vol. 1**
A. Palaci, H. Thiriez - **Multiplan per Apple II plus e Apple IIe**
J. Deconchat - **102 programmi per MSX**
J.-F. Séhan - **MSX in famiglia**
J. Boisgontier - **Basic MSX. Metodi pratici**
C. Bardon, B. De Merly - **Giochi sul Philips C7420 Videopac +**
V.C. Grandis, J. Deconchat - **102 programmi per Philips C7420 Videopac +**
J.P. Richard - **La scoperta del PC 1500**
J.-L. Marx, A. Thibault - **Chiavi per Lotus 1-2-3**
A. Pinaud - **Programmare in Forth**
A. Pinaud - **CP/M passo dopo passo**
Y. Dargery - **Memento per MS-DOS**
A. Pinaud - **MS-DOS passo dopo passo**
M. Thibault - **Chiavi per MULTIPLAN**
J.-P. Blanger - **Modelli di espressione grafica**
C. Nowakowski - **Programmare in C**

Volumi di prossima pubblicazione

N. Bréaud-Pouliquen - **Chiavi per Apple IIc**
H. Thiriez - **Multiplan per IBM PC**
H. Thiriez, G. Santraille - **Multiplan per Macintosh**
J. Deconchat - **102 Programmi per AMSTRAD**
J. Boisgontier, B. Cesard - **Basic AMSTRAD**
N. Bréaud-Pouliquen - **Le risorse dell'Apple IIc**
J. Rouault, P. Girard - **Pascal UCSD per Apple - Vol. 2**
F. Verscheure - **PRODOS su Apple IIe e Apple IIc**
P. Brandeis, F. Blanc - **Assembler e periferiche MSX**
J.P. Richard - **La scoperta del PC 1251**
D.A. Lien - **Dizionario del Basic IBM**

JACQUES DECONCHAT

Adattamento di
ALAIN LAVENIR

102 programmi per ATARI

600 XL, 800 XL, 130 XE

Traduzione di
ANDREA PANAGIA

Editsi - Editoriale per le scienze informatiche - S.r.l.
MILANO 1985

Sommario

Si ringrazia per la collaborazione prestata l'ATARI ITALIA S.p.A. di Cinisello Balsamo (MI)

Prefazione

Troverete in questo libro più di cento programmi (essenzialmente giochi), redatti in Basic Atari per i computer ATARI. La maggior parte di questi programmi saranno d'altronde utilizzabili senza grosse modifiche sugli altri calcolatori che utilizzano un Basic come linguaggio di programmazione.
I programmi proposti sono molto corti e l'ordine nel quale sono presentati non è certamente arbitrario: si tratta di iniziare il lettore, tramite la realizzazione e l'utilizzazione dei programmi di giochi di buona qualità, alla conoscenza e alla padronanza progressiva di questo strumento straordinario che è un personal computer.

La brevità dei programmi proposti permetterà di batterli facilmente, senza il rischio di commettere degli errori. Potrete naturalmente utilizzarli come base per la realizzazione di programmi più impegnativi.

I programmi sono abbondantemente commentati per aiutarvi a capire il loro funzionamento e le pagine di presentazione vi permetteranno di ritrovare rapidamente le caratteristiche proprie delle istruzioni Basic che utilizzano.

I programmi non utilizzano tutte le possibilità di un apparecchio come l'Atari.

Il livello 5 dà un'idea sull'utilizzazione di alcune tecniche abbastanza particolari che potranno essere riprese con profitto per migliorare i programmi precedenti.

Troverete, alla fine di ogni presentazione, qualche suggerimento per possibili migliorie, o per modifiche che potrebbero rendere il gioco proposto più attraente, più dinamico o più completo.

Organizzazione dell'opera

I giochi sono classificati in ordine crescente di difficoltà seguendo 5 livelli:

- **Il livello 1** permette lo studio delle istruzioni elementari del linguaggio Basic: **LET** (opzionale) **PRINT, INPUT, IF. . . THEN, FOR. . . NEXT, GOTO, GOSUB, ON. . . GOTO, ON. . . GOSUB**.
- **Al livello 2**, si utilizzeranno le istruzioni di gestione e di manipolazione degli elementi di un array e la ricerca di un elemento in una stringa: **DIM, R(I), READ, DATA, RESTORE**.
- **Al livello 3** si utilizzerà la serie completa delle istruzioni di utilizzo di stringhe di caratteri e le istruzioni che permettono la visualizzazione e la grafica con differenti livelli di risoluzione: **COLOR, PLOT, GRAPHICS, DRAWTO**.
- **Il livello 4** tratta un'istruzione molto particolare, che facilita la realizzazione di giochi interattivi: **GET** ed esplora le risorse di **POKE**.
- Per finire, **il livello 5** ci farà scoprire le istruzioni più specifiche: **LOCATE, TRAP, POP** e costituisce il risultato dell'opera.

Ogni gioco è presentato in maniera sistematica con una corta descrizione introduttiva, seguita dallo studio dettagliato di ognuna delle linee importanti del programma. Seguono le indicazioni necessarie al buon funzionamento del gioco e alcuni suggerimenti per estendere o migliorare il gioco proposto. Segue, infine, il listato del programma funzionante su ATARI, ottenuto su una stampante ATARI per garantire contro rischi di malfunzionamento.

LIVELLO 1

N.	Nome del programma	Tema e particolarità
1	Il numero misterioso	Gioco di deduzione, per 1 giocatore. Vari livelli
2	Cubo	Gioco di disegno: prospettiva sullo schermo
3	Caldo, freddo	Logica e fortuna, per 1 giocatore. Vari livelli
4	Fiammiferi	Gioco di riflessione contro il calcolatore
5	Fiammiferi in molti	Gioco di riflessione e di strategia, vari giocatori
6	Calendario perpetuo	Creazione di un calendario mese per mese
7	Battaglia navale	Una sola nave da trovare, con qualche particolarità
8	Golf	Un percorso con 18 buche, per 1 solo giocatore
9	Solfeggio	Imparate il solfeggio con l'aiuto del calcolatore
10	Blackjack	Gioco di fortuna, il calcolatore tiene banco
11	Pari-Dispari	Un gioco da Casinò in tre tempi per 1 o più giocatori
12	Indovina un numero	Un gioco molto facile, per 1 o più giocatori
13	Il gioco di Wytthof	Un gioco di riflessione abbastanza semplice, per due giocatori
14	Rimbalzi	Un gioco di abilità molto visivo, per 1 giocatore
15	Jackpot	Un gioco di fortuna, per 1 giocatore
16	Roulette	Una simulazione elegante del famosissimo gioco
17	Corridoi	Un gioco di logica da scoprire
18	Scacchiera	Una base grafica per le vostre creazioni
19	Il gioco della fava	Un gioco di riflessione, per 2 giocatori
20	Gioco dell'oca casuale	Un vecchissimo gioco, rivisto e corretto dal calcolatore, per 2 giocatori.

Istruzioni elementari

Tratteremo qui solo le istruzioni fondamentali, o istruzioni di programmazione; molti altri comandi, o funzioni particolari, saranno utilizzati quando occorre. Per esempio:

— ***RND(1) per ottenere un numero casuale tra 0 e 1.***

— ***INT per prendere la parte intera di un risultato.***

— ***REM per mettere un commento.***

— ***END per terminare un programma.***

— ***DIM per riservare uno spazio a una stringa di caratteri.***

Le istruzioni fondamentali della programmazione, utilizzate già al livello 1 e presenti praticamente in tutti i programmi sono:

LET (opzionale); **PRINT; INPUT; GOTO|___|; IF|___|THEN|___|; FOR. . .NEXT; GOSUB. . .RETURN.**

ossia solamente sette istruzioni di base (e due istruzioni derivate) che esamineremo ora più dettagliatamente.

N.B. Tutte le istruzioni hanno una forma abbreviata che facilita la battitura: bastano generalmente le prime due lettere dell'istruzione, seguite da un punto: per esempio PR. = PRINT.

L'ISTRUZIONE LET, O ISTRUZIONE DI ASSEGNAZIONE

È l'istruzione per eccellenza dei linguaggi di programmazione evoluti, visto che il suo ruolo essenziale è di permettere, per il programmatore, una gestione trasparente della memoria. Questo significa, in pratica, che per inserire una quantità in memoria, per esempio 12, non è necessario sapere in quale locazione di memoria la si deporrà: basta assegnarle un nome autorizzato, col quale la quantità sarà in seguito ritrovata e manipolata.

— **Se la quantità è numerica,** il nome sarà una semplice lettera dell'alfabeto, o un insieme di varie cifre o lettere, che inizia obbligatoriamente con una lettera.

— **Se la quantità è alfanumerica,** si applica la stessa regola, ma il nome dovrà essere seguito dal simbolo $ e la variabile dovrà essere scritta tra virgolette.

Esempio:

LET A1=12; la variabile numerica A1 prende il valore 12.
LET NOM$="RENOIR"; la variabile alfanumerica NOM$ prende il valore RENOIR.

N.B. Prima bisognava dimensionare la variabile NOM$ con il comando DIM. Così per esempio il comando DIM NOM$(50) limita la stringa NOM$ a un massimo di 50 caratteri.

In ATARI, come nella maggior parte dei Basic, il LET è diventato opzionale, non essendo possibile nessuna ambiguità, scriveremo dunque: A=12 ; NOM$="TOTO"; ecc.

L'ISTRUZIONE PRINT O ISTRUZIONE DI USCITA (SCHERMO)

È un'istruzione di dialogo, che permette al calcolatore di visualizzare sullo schermo un testo, il risultato di un calcolo, o una variabile. È il solo modo che possiede il personal computer per far conoscere i suoi risultati. Così PRINT A1 provoca la visualizzazione di 12 (se era stato fatto prima A1=12) ; PRINT A$ provoca la visualizzazione di BUONGIORNO (se era stato fatto prima A$="BUONGIORNO"). Utilizzato dopo aver cancellato lo schermo, il PRINT provoca la visualizzazione sulla prima linea, in alto, a sinistra.
Un certo numero di parametri possono essere utilizzati per ottenere sullo schermo una visualizzazione adeguata:

— Un ; **(punto e virgola)** a seguito del PRINT permette sia di concatenare un'altra stampa senza ripetere l'istruzione PRINT, sia di indicare che la prossima stampa sarà, in assenza di un'indicazione contraria, effettuata esattamente al seguito della stampa precedente.

— La , **(virgola)** provoca una tabulazione orizzontale automatica predefinita.

Durante la scrittura, **l'istruzione PRINT** potrà essere abbreviata battendo **? (punto interrogativo)**. Durante una stampa, ogni numero è preceduto da un segno ; ma il segno positivo non è visualizzato.

L'ISTRUZIONE POSITION

L'ATARI dispone di una istruzione particolare, di uso molto elastico, che ***permette di posizionare il cursore sullo schermo, in vista della prossima scrittura.*** Si tratta dell'istruzione POSITION X,Y. X corrisponde ad una tabulazione orizzontale (tra 0 e 39) e Y ad una tabulazione verticale (tra 0 e 23).
Così, per scrivere la parola BUONGIORNO alla quinta colonna dell'ottava riga, si scriverà:

```
POSITION 5,8;PRINT "BUONGIORNO"
```

N.B. senza l'istruzione POSITION, gli inizi di riga cominciano alla 2ª colonna.

L'ISTRUZIONE INPUT, O ISTRUZIONE D'INGRESSO (TASTIERA)

Questa istruzione, che provoca l'arresto e l'attesa del calcolatore, ***permetterà di inserire un numero, o un testo, che sarà assegnato alla variabile attraverso l'ordine INPUT.*** Questo è fatto in modo conversazionale, cioè in funzione delle necessità; infatti, quando l'ATARI trova una istruzione INPUT, si ferma e vi passa la mano; si può allora inserire un'informazione di natura adeguata (numero o testo) e segnalare che questo inserimento è terminato con il comando **RETURN**, il che gli restituirà la mano.
Se l'informazione inserita non è di natura adeguata (battete, per esempio, un testo su un INPUT X, che aspetta un numero), sarà segnalato errore con arresto del programma.

L'ISTRUZIONE GOTO O ISTRUZIONE DI SALTO INCONDIZIONATO

Una delle particolarità del linguaggio Basic, che lo rende facilmente accessibile, è che ***ogni linea di programma deve essere preceduta da un numero***, queste linee sono in seguito eseguite dal personal computer in sequenza, nel senso crescente dei numeri. Per diverse ragioni, potrebbe essere necessario durante lo svolgimento di un programma, non seguire questo ordine sequenziale molto rigido; utilizzeremo allora ***l'istruzione GOTO, che permette di dirottare il funzionamento del programma dal suo corso normale***, con la possibilità d'altronde di riprenderlo in seguito con un altro GOTO.
Il rinvio deve peraltro essere fatto a un numero di riga esistente nel programma (GOTO 20, per esempio).
Per dare un po' di elasticità a questa istruzione di rinvio, esiste un'istruzione molto potente, che permette di rinviare ad un qualunque numero di linea, secondo il valore di una variabile numerica: **è l'istruzione ON|____|GOTO|____|**; scriveremo, per esempio:

ON X GOTO 100, 200, 300

ed il rinvio sarà fatto alla linea 100 se X vale 1, alla linea 200 se X vale 2, e 300 se X vale 3.

L'ISTRUZIONE IF|____|THEN|____| O ISTRUZIONE DI TEST

Possiamo utilizzarla in due maniere diverse. In rottura di sequenza condizionale sotto la forma:

IF (condizione = vera) THEN (numero di linea).

È l'utilizzo più corrente di questo test, che permette di rinviare ad un'altra parte del programma, se per esempio X=0, o se A=B, ecc. **Gli operatori logici NOT, OR** o **AND** possono d'altronde essere utilizzati nella scrittura della condizione.

Si puo però utilizzare il test nella forma:

IF (condizione) THEN (istruzione) ;

l'istruzione presentata al seguito di THEN può essere un'assegnazione: es. A=12, una stampa: es. PRINT A, una immissione: es. INPUT B$, o anche un altro test IF|___|THEN|___|.

Possiamo per esempio scrivere:

```
IF A>3 THEN IF B<7 THEN PRINT A+B
```

L'ISTRUZIONE FOR...NEXT, O ISTRUZIONE DI CICLO

Siamo spesso condotti, in programmazione, a voler eseguire un numero intero di volte una certa serie d'istruzioni (per esempio, fare avanzare un punto mobile dalla casella 6 alla casella 20, ecc.). Piuttosto che utilizzare un test ed un rinvio, sarà meglio utilizzare un ***ciclo***; la variabile di ciclo sceglie i suoi valori negli interi o nei decimali e il ***passo*** di questa variabile (cioè la quantità di cui aumenta o diminuisce ad ogni passaggio) può anch'esso scegliere i suoi valori negli interi o nei decimali.

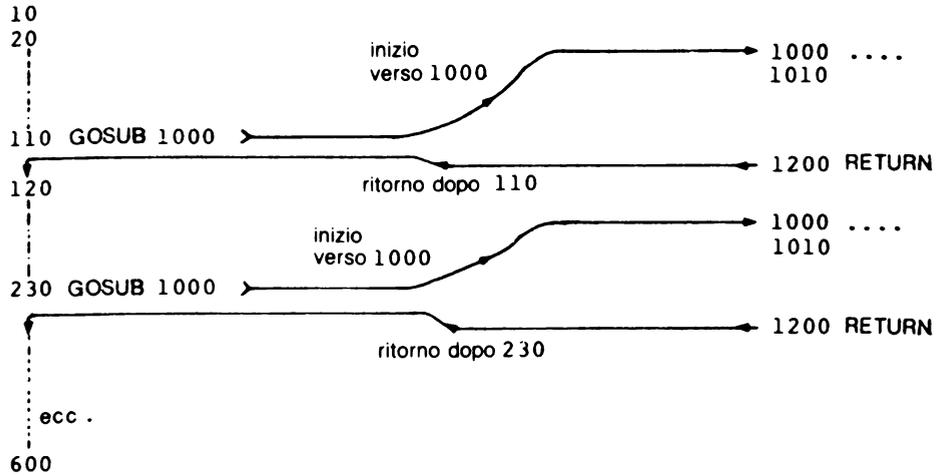
Così se scriviamo:

```
FOR I =  ⎫ 0 TO 20 STEP 2
         ⎬ istruzioni
NEXT I   ⎭
```

Le istruzioni contenute nel ciclo saranno eseguite per I=0, per I=2,... e questo fino a I=20, ossia 11 volte. La variabile I può essere usata per fare un calcolo nel ciclo, ma non dovrà esservi modificata. Possiamo, naturalmente, uscire dal ciclo prima di I=20, con un test condizionale indicante una quantità qualunque.
Se ***il passo del ciclo (STEP)*** non è precisato, sarà automaticamente preso uguale a 1.
È possibile concatenare vari cicli di nome diverso.

L'ISTRUZIONE GOSUB, O ISTRUZIONE DI RICHIAMO DI SOTTOPROGRAMMA

Quando una stessa serie di istruzioni deve essere eseguita varie volte, e richiamata da diverse zone del programma, eviteremo di riscriverla più volte. La scriveremo una volta per tutte e la posizioneremo fuori del programma vero e proprio (dandole, per esempio, un numero di linea superiore al numero dell'ultima linea del programma). Indipendentemente dalle difficoltà che potremo incontrare per i nomi delle variabili, quando vorremo utilizzare questa sequenza, basterà chiamarla con **l'istruzione GOSUB (numero di linea)**.
Il programma farà allora una deviazione nella parte richiamata, ma non potrà proseguire il suo svolgimento normale se non incontra un'istruzione **RETURN** o istruzione di ritorno.

Ecco lo schema teorico di funzionamento:

Un sottoprogramma potrà a sua volta eventualmente richiamarne un altro e così via.
Possiamo anche richiamare un sottoprogramma in modo condizionato, usando l'istruzione **ON|___|GOSUB|___|.**

Per esempio ON X GOSUB 1000, 2000, 3000 permetterà di eseguire il sottoprogramma posizionato a 1000 se X=1, a 2000 se X=2, a 3000 se X=3.

N.B. Per facilitare l'accesso al programma è possibile utilizzare un segnale (cioè un nome alfabetico) per rappresentare un numero di passo del programma dato. Così, se MUSIQUE = 1000, l'istruzione "GOTO MUSIQUE" equivale a "GOTO 1000".

Il numero misterioso

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di deduzione per un giocatore.
Difficoltà del gioco: facile.

IL GIOCO: Si chiede al calcolatore di definire un numero di n cifre (con n compreso tra 1 e 7).
Si dovrà quindi cercare di indovinarlo col minor numero di tentativi possibile, ricevendo le sole informazioni TROPPO GRANDE o TROPPO PICCOLO.

IL PROGRAMMA: La linea 70 permette di creare il numero da indovinare.
Gli inserimenti sono fatti dalle linee 90-100.
La linea 110 verifica se il risultato è esatto.
La linea 120 segnala un tentativo troppo piccolo e la linea 130 uno troppo grande.
Il contatore di tentativi è alla linea 140.
Le linee da 200 a 230 si occupano della visualizzazione dei risultati e della gestione della parte seguente.

PER GIOCARE: Si farà partire il programma con RUN.
Si otterrà un primo messaggio NUMERO DI CIFRE? al quale si risponde con un intero tra 1 e 7, seguito da RETURN.
Si avrà allora il messaggio PROVA N.1, seguito dalla domanda IL TUO NUMERO.
Si inserisce un numero e si schiaccia RETURN.
Si ottiene allora la risposta TROPPO PICCOLO o TROPPO GRANDE, oppure, se si è riusciti ad indovinare, il messaggio HAI TROVATO....IN....PROVE.
Se il risultato non è esatto, si ritorna a PROVA N.2, e IL TUO NUMERO.
Alla fine del gioco, appare la domanda UN'ALTRA PARTITA? (S/N); si risponde S per fare un'altra partita (seguito da RETURN).
Si può fermare il gioco in qualsiasi momento battendo BREAK.

ESTENSIONI POSSIBILI: Potremmo immaginare di fare indovinare al giocatore non un numero ma qualcos'altro (per esempio una parola, utilizzando l'ordine alfabetico, o un'altra cosa ancora...).
Nota per rientrare nel programma: il carattere ↰ si ottiene con ESC, SHIFT, CLEAR

```
10 REM IL NUMERO MISTERIOSO
20 DIM R$(1)
30 PRINT "}"
40 PRINT "NUMERO DI CIFRE:";:INPUT N
50 IF N>7 THEN RUN
60 IF N<=0 THEN RUN
70 X=INT(RND(1)*10^N+1)
80 J=1
90 PRINT :PRINT "PROVA No ";J
100 PRINT "IL TUO NUMERO :";:INPUT M
110 IF M=X THEN 200
120 IF M<X THEN PRINT "TROPPO PICCOLO"
130 IF M>X THEN PRINT "TROPPO GRANDE"
140 J=J+1
150 PRINT :PRINT
160 GOTO 90
200 PRINT "HAI TROVATO ";X;" IN ";J;" PROVE"
220 PRINT "UN'ALTRA PARTITA (S/N)";:INPUT R$
230 IF R$="S" THEN RUN
```

Cubo

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma si propone di realizzare con l'aiuto dell'istruzione PRINT una vista di un cubo in prospettiva.
Potrebbe essere per esempio incorporato in un gioco basato sulle capacità di riconoscimento delle forme geometriche elementari.

IL PROGRAMMA: Il disegno del cubo è realizzato mediante punti, utilizzando due cicli: il ciclo I (30-120) e il ciclo I (130-180).
Il nome dell'oggetto raffigurato è scritto alla linea 200 (qui: CUBO).

PER GIOCARE: Basta far partire il programma con RUN e guardare.

ESTENSIONI POSSIBILI: L'alta risoluzione permetterà di avere un tracciato più preciso.
Potremmo immaginare un gioco utilizzante dei parallelepipedi di differenti misure, con visualizzazione per esempio dei tre lati, dovendo calcolare il volume, ecc.

```
10 REM CUBO
20 GRAPHICS 0
30 FOR I=0 TO 10
40 POSITION 7+I,0:PRINT "."
50 POSITION I,7:PRINT "."
60 POSITION 7+I,10:PRINT "."
70 POSITION I,17:PRINT "."
80 POSITION 0,7+I:PRINT "."
90 POSITION 10,7+I:PRINT "."
100 POSITION 7,I:PRINT "."
110 POSITION 17,I:PRINT "."
120 NEXT I
130 FOR I=0 TO 7
140 POSITION I,7-I:PRINT "."
150 POSITION 10+I,7-I:PRINT "."
160 POSITION I,17-I:PRINT "."
170 POSITION 10+I,17-I:PRINT "."
180 NEXT I
190 POSITION 26,19:PRINT "";
200 POSITION 26,20:PRINT "CUBO";
210 POSITION 26,21:PRINT "";
```

Caldo, freddo

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Uno dei più vecchi giochi del mondo, sempre in voga, ma questa volta è il calcolatore che vi fa da compagno: infatti nasconderà un oggetto in una certa posizione di una griglia di cui avrete voi stessi scelto le dimensioni. Per trovarlo, non vi darà che qualche indicazione, sotto la forma BRUCI, o FA FREDDO, ecc. Dovete riuscire a trovare l'oggetto nascosto col minor numero di tentativi possibile (il vostro punteggio sarà visualizzato alla fine della partita).

IL PROGRAMMA: Le linee 30 e 40 definiscono la posizione dell'oggetto.
La linea 90 calcola la distanza in funzione delle due coordinate che avete inserito alle linee 60 e 70.
La linea 100 controlla se avete vinto.
Le linee 120-130 normalizzano la distanza (che deve avere un senso, quali siano le dimensioni della griglia iniziale).
La linea 140 effettua la partenza, se c'è posto, verso uno dei sottoprogrammi di visualizzazione.

PER GIOCARE: Si fa partire il programma con RUN, poi si inseriscono le dimensioni della griglia (quadrata) e si batte RETURN.
Già dalla visualizzazione seguente FORNISCI X, si inserisce la coordinata supposta di X, seguita da RETURN.
Si ottiene in seguito il messaggio FORNISCI Y e si inserisce Y, seguito da RETURN.
Si ottiene in risposta un testo, che varia in funzione della distanza dall'oggetto cercato (si va da COMPLETAMENTE GELATO a BRUCIATO). Quando lo si è trovato, si ottiene la visualizzazione del numero di tentativi effettuati.
Per uscire dal programma prima della fine della partita, basta premere BREAK.

ESTENSIONI POSSIBILI: Se si dispone del colore, sarà interessante trovare un modo più immediato per simbolizzare la distanza, per esempio delle mattonelle varianti dal rosso al blu nelle posizioni tentate.
Si potrebbe anche lavorare in uno spazio a tre dimensioni, nascondere vari oggetti ecc.

```
10 REM CALDO,FREDDO
20 PRINT "DIMENSIONE DELLA GRIGLIA :";:INPUT D
30 X=INT(RND(1)*D+1)
40 Y=INT(RND(1)*D+1)
50 PRINT "}"
60 PRINT "Fornisci X :";:INPUT A
70 PRINT "fornisci Y :";:INPUT B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THEN 6000
110 PRINT "Sei ";
120 N=INT(SQR(D)/T)+1
130 IF N>5 THEN N=5
140 IF T<=SQR(D) THEN ON N GOTO 1000,2000,3000,4000,5000
150 PRINT "completamente GELATO."
160 GOTO 60
1000 PRINT "appena TIEPIDO."
1010 GOTO 60
2000 PRINT "TIEPIDO."
2010 GOTO 60
3000 PRINT "CALDO."
3010 GOTO 60
4000 PRINT "MOLTO CALDO."
4010 GOTO 60
5000 PRINT "BRUCIATO."
5010 GOTO 60
6000 PRINT "}"
6010 PRINT "trovato in ";S;" prove"
```

Fiammiferi

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: contro l'ATARI.
Difficoltà del gioco: media.

IL GIOCO: Il giocatore dovrà all'inizio della partita scegliere un certo numero di fiammiferi, così come la presa massima autorizzata. Decide quindi se cominciare, o se lasciare la prima mossa all'ATARI. Ognuno prende a turno tanti fiammiferi quanti ne desidera, nel limite del massimo consentito.
Quello che prende l'ultimo fiammifero è considerato perdente.

IL PROGRAMMA: La mano del giocatore comincia alla linea 70.
Quella dell'ATARI alla linea 110.
La linea 80 controlla la legalità della giocata del giocatore.
La linea 90 gestisce il cambiamento del giocatore.
La mano dell'ATARI è calcolata in un ciclo non formale, dalla linea 120 alla linea 150.
Il risultato è visualizzato dalla linea 1010.
Se la partita non è terminata, la linea 1030 lo percepisce.

PER GIOCARE: Si parte con RUN e, all'apparizione del messaggio NUMERO TOTALE DI FIAMMIFERI, si inserisce il numero di fiammiferi, seguito da RETURN. Poi si inserisce la presa massima autorizzata, seguita da RETURN. Se il giocatore vuole iniziare, batte in seguito 0; altrimenti batte 1, poi RETURN.
Dopo ogni mano è visualizzato il numero di fiammiferi rimanenti e l'ATARI avverte TIRA ANCORA quando tocca al giocatore prendere.
Chi prende l'ultimo fiammifero ha perso.

ESTENSIONI POSSIBILI: Potremmo naturalmente immaginare una visualizzazione grafica a colori, potremmo anche lasciare al caso la scelta del numero di fiammiferi e la presa massima, ecc.

```
10 REM FIAMMIFERI
20 GRAPHICS 0
30 PRINT "Numero totale di fiammiferi    :";:INPUT N
40 PRINT "Presa massima per colpo        :";:INPUT I
50 PRINT "Chi comincia (Tu:0- Io:1)      :";:INPUT D
60 IF D=1 THEN 110
70 PRINT "Tira ancora  :";:INPUT P
80 IF P<>INT(P) OR P<1 OR P>I OR P>N THEN 70
90 D=1
100 GOTO 1000
110 S=I+1
120 T=(N-S)/(I+1)
130 IF T=INT(T) THEN 200
140 S=S-1
150 GOTO 120
200 P=ABS(S-1)
210 IF P=0 THEN P=1
220 D=0
230 PRINT "Prendo ";P
1000 N=N-P
1010 PRINT "Rimansono ";N;" fiammiferi."
1020 PRINT
1030 IF N<>0 THEN 60
1040 PRINT "}"
1050 IF D=1 THEN PRINT "Ho vinto..."
1060 IF D<>1 THEN PRINT "Bravo"
```

Fiammiferi in molti

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: logica (vari giocatori).
Difficoltà del gioco: elevata.

IL GIOCO: In questo gioco per vari giocatori, il direttore del gioco dovrà, all'inizio della partita, scegliere un numero di fiammiferi ed indicare all'ATARI il numero di giocatori.
Ad ogni mano il giocatore di turno è autorizzato a prendere dal mucchio di fiammiferi rimanenti fino al doppio di quanto ha preso il giocatore precedente; colui che prende l'ultimo fiammifero ha vinto (o perso se si modificano leggermente le regole).

IL PROGRAMMA: Le linee da 10 a 40 permettono i differenti ingressi.
Le linee da 70 a 100 realizzano le stampe.
La linea 120 controlla la mano di ogni giocatore (in rapporto alla mano precedente).
La linea 170 gestisce le mani (rimessa a 1 del numero del giocatore).
La linea 150 realizza il test finale e la linea 200 visualizza il numero del giocatore vincente (con la regola prevista inizialmente).

PER GIOCARE: Dopo aver lanciato il programma con RUN, si inserisce il numero di fiammiferi iniziale, seguito da RETURN, poi il numero di giocatori (e RETURN).
Il primo giocatore potrà prendere solo un fiammifero, il secondo ne potrà prendere uno o due; il seguente 1,2, o 1,2,3 o 4 in funzione della mano precedente, e così via; il numero massimo è ogni volta visualizzato dall'ATARI. Quando l'ultimo fiammifero è stato preso si ottiene il messaggio IL GIOCATORE . . . HA VINTO e la partita è terminata.

ESTENSIONI POSSIBILI: Si può decidere che quello che prende l'ultimo fiammifero è eliminato dal gioco e ricominciare la partita con i giocatori rimanenti che saranno anch'essi eliminati gli uni dopo gli altri.
Possiamo anche immaginare una rappresentazione grafica del mucchio di fiammiferi, scegliere a caso il numero da prendere all'inizio della partita ecc.

```
10 REM FIAMMIFERI IN MOLTI
20 GRAPHICS 0
30 PRINT "Numero totale di fiammiferi  :";:INPUT N
40 PRINT "Numero di giocatori          :";:INPUT J
50 F=1
60 K=1
70 PRINT "}"
80 PRINT "Turno del giocatore ";K
90 PRINT "Totale rimasto     :";N
100 PRINT "Massimo da togliere :";F
110 PRINT "Il tuo colpo :";:INPUT C
120 IF C=0 OR C>F OR C>N THEN 110
130 N=N-C
140 F=2*C
150 IF N=0 THEN 190
160 K=K+1
170 IF K>J THEN K=1
180 GOTO 70
190 PRINT "}"
200 PRINT "Il giocatore ";K;" ha vinto."
```

Calendario perpetuo

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: gadget pubblicitario.
Difficoltà del gioco: nessuna.

IL GIOCO: Questo programma vi permetterà di creare tutti gli anni il vostro calendario; potrete così servirvene per mandare ai vostri amici un calendario del mese della loro nascita, o per augurar loro buone feste, ecc.

IL PROGRAMMA: Le linee 30 e 40 vi permettono di inserire il mese e l'anno.
Le linee 50, 60 e 70 calcolano il numero del primo giorno del mese.
La linea 80 calcola il numero del giorno del mese (non si è tenuto conto qui del 29 febbraio, per semplificare).
Il ciclo A (130-160) realizza la stampa del calendario.

PER GIOCARE: Dopo aver lanciato il programma con l'istruzione RUN, si dovrà introdurre l'anno, per esempio 1985, seguito da RETURN.
Si ottiene allora la domanda MESE, alla quale conviene rispondere con il numero del mese, secondo l'ordine usuale (1 per GENNAIO, 2 per FEBBRAIO, ecc.).
Si batte in seguito RETURN; allora il calendario viene visualizzato sullo schermo.

ESTENSIONI POSSIBILI: Potremmo prevedere senza difficoltà la stampa dei 29 Febbraio, prevedere un'opzione che permetta la stampa per un anno completo, prevedere l'introduzione del nome dei mesi in chiaro, ecc.

```
10 REM CALENDARIO PERPETUO
20 GRAPHICS 0
30 PRINT "Quale anno   :";:INPUT A
40 PRINT "Quale mese   :";:INPUT M
50 D=INT(365.25*(A-(M<3)))+1
60 D=D+INT(30.6*(M+1+(M<3)*12))-INT((INT((A-(M<3))/100)-7)*0.75)
70 D=D-7*INT(D/7)
80 M=31-(M=4 OR M=6 OR M=9 OR M=11)-(M=2)*3
90 PRINT
100 PRINT "DOM LUN MAR MER GIO VEN SAB"
110 PRINT
120 L=6
130 FOR A=1 TO M
140 POSITION 3+4*(D+A-1),L:PRINT A
150 IF D+A-1=6 THEN L=L+1:D=D-7
160 NEXT A
```

Battaglia navale

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessione.
Difficoltà del gioco: bassa.

IL GIOCO: Una battaglia navale a priori abbastanza semplice, con una sola nave nascosta a caso in una griglia di 10 * 10 caselle.
Per ogni colpo bisogna indicare le coordinate O (orizzontale) e V (verticale) della casella mirata; l'ATARI risponde visualizzando la distanza che separa la nave invisibile dalla casella mirata; se si colpisce troppo vicino, la nave può nascondersi nuovamente in modo casuale.

IL PROGRAMMA: Il ciclo I (40-70) disegna il campo di gioco.
Le linee 90-100 definiscono la posizione della nave.
Le coordinate di tiro sono inserite alle linee 120 e 130; la linea 140 visualizza il colpo sullo schermo. La distanza è calcolata alla linea 150; la linea 160 verifica se si è fatto o no centro.
La linea 180 interviene per spostare la nave se i colpi si avvicinano troppo.
La linea 190 permette la visualizzazione della distanza.

PER GIOCARE: Il campo di gioco è visualizzato dopo aver battuto RUN. Al messaggio RIGA, si risponde dando il numero della linea orizzontale sulla quale si desidera tirare. Si batte quindi RETURN e alla domanda COLONNA si risponde dando il numero della colonna verticale seguito da RETURN.
Se si tira troppo vicino, si ottiene il messaggio PFFFF... che significa che la nave nemica ha cambiato posizione.
Se si tira sulla casella giusta, si ottiene la visualizzazione del numero totale dei tiri effettuati per vincere.

ESTENSIONI POSSIBILI: Sono innumerevoli, anche perché la battaglia navale è uno dei giochi più famosi che esistano. Possiamo introdurre due giocatori, giocare contro l'ATARI, mettere varie navi, dei sottomarini, ecc.

```
10 REM BATTAGLIA NAVALE
20 GRAPHICS 0
30 PRINT "   0  1  2  3  4  5  6  7  8  9"
40 FOR I=0 TO 9
50 PRINT I;"  .  .  .  .  .  .  .  .  .  ."
60 PRINT
70 NEXT I
80 S=S+1
90 X=INT(RND(1)*10)
100 Y=INT(RND(1)*10)
110 POSITION 0,21:PRINT "                                       "
120 POSITION 0,21:PRINT "Riga :";:INPUT H
130 POSITION 20,21:PRINT "Colonna :";:INPUT V
140 POSITION 3*V+5,2*H+1:PRINT S
150 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
160 IF D=0 THEN 200
170 S=S+1
180 IF D=1 THEN POSITION 0,22:PRINT "PFFFF...  ":GOTO
 90
190 POSITION 0,22:PRINT "Distanza     :";D;"   "
200 GOTO 110
210 PRINT "}"
220 PRINT "TROVATO IN ";S;" COLPI"
```

Golf

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Si tratta di riuscire a fare, col minor numero di colpi possibile, un percorso di golf con 18 buche. Per fare ciò, occorre colpire la palla con una forza più o meno grande, fino ad ottenere la sua caduta nella buca.
Ma attenzione, il caso potrebbe intervenire su ogni colpo, così spesso dovrete tornare indietro (bisognerà allora far precedere l'indicazione della forza con un segno —).

IL PROGRAMMA: La linea 50 permette di definire la posizione della buca e la linea 60 disegna il campo di gioco.
Lo spostamento da realizzare è calcolato alla linea 100.
Il ciclo I (120-170) realizza lo spostamento della palla.
La linea 180 guarda se il colpo è, o no, vincente; se lo è, si ricomincia (linea 230) fino a che si realizzino 18 buche (ciclo S (20-230); fine del ciclo in 250, con visualizzazione di fine).

PER GIOCARE: Si ottiene la visualizzazione del campo con RUN così come il disegno della palla nella sua posizione iniziale ed il testo: BUCA 1. FORZA; si deve allora rispondere dando la forza desiderata (un numero tra 1 e 300...).
Subito dopo aver battuto RETURN, la palla ricomincia a spostarsi; può a questo punto sia fermarsi davanti alla buca (nel qual caso si ricomincia), sia finirci dentro (si passa allora alla buca seguente), sia sorpassarla: si dovrà allora batterla con una forza negativa.
Il gioco finisce con la scritta 18 BUCHE IN . . . COLPI.

ESTENSIONI POSSIBILI: Potremmo espandere il gioco per fare partecipare più giocatori, migliorare la qualità grafica, utilizzare una rappresentazione spaziale del campo di gioco riferirsi ad alcuni videogiochi), avere un terreno più accidentato, degli ostacoli, cambiare il club...

```
10 REM GOLF
20 FOR S=1 TO 18
30 GRAPHICS 0
35 X=1
40 X=1
50 T=INT(29*RND(1)+9)
60 POSITION 1,6:PRINT ""
70 POSITION T,6:PRINT " ":POSITION X,5:PRINT ""
80 POSITION 15,8:PRINT "        "
90 POSITION 0,8:PRINT "Buca ";S;"    Forza :";:INPUT R
100 F=INT(R*0.16-(ABS(R)*SQR(RND(1)+0.1)^2)/1000)
110 IF F=0 THEN 70
120 FOR I=SGN(F) TO F STEP SGN(F)
130 X=X+SGN(F)
140 IF X<39 AND X>=1 THEN POSITION X-SGN(F),5:PRINT " ":POSITION X,5:PRINT ""
150 IF X>=38 THEN X=38
160 IF X<=1 THEN X=1
170 NEXT I
180 IF X=T THEN 210
190 E=E+1
200 GOTO 60
210 POSITION X,5:PRINT " ":POSITION X,6:PRINT ""
220 FOR P=1 TO 500:NEXT P
230 NEXT S
240 PRINT "}"
250 PRINT "18 buche in ";E;" colpi."
```

Solfeggio

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessione.
Difficoltà del gioco: bassa.

IL GIOCO: Questo programma vi propone un'iniziazione elementare al solfeggio, permettendovi di imparare a riconoscere facilmente le differenti note, che appaiono le une dopo le altre su uno spartito; dopo la visualizzazione della nota, dovete inserire il suo nome in chiaro ed il vostro punteggio sarà visualizzato in permanenza.
Il programma è previsto inizialmente per la chiave di sol, ma sarà facilmente adattabile ad un'altra chiave.

IL PROGRAMMA: Il ciclo I (50-70) disegna uno spartito.
La nota è scelta alla linea 80 e visualizzata alle linee 90-100-110-120.
La linea 130 visualizza il punteggio.
La linea 140 visualizza il nome della chiave (qui chiave di sol) e vi interroga.
La linea 160 si occupa della fine della partita.
Le linee 170 e 180 effettuano la decodificazione della nota inserita.
La linea 200 controlla il risultato.

PER GIOCARE: Dopo aver lanciato il programma con RUN, apparirà lo spartito assieme alla prima nota.
Risponderete alla domanda QUAL È LA NOTA? dandone il nome in lettere (do, re, mi, fa, sol, la, si), seguito da RETURN.
Il punteggio vi indicherà se è esatto o no, ed una nuova nota sarà visualizzata.
Per fermare, basterà rispondere 0 seguito da RETURN, alla domanda QUAL È LA NOTA?

ESTENSIONI POSSIBILI: Potremmo innanzitutto facilmente modificare il programma per permettervi di lavorare con le diverse chiavi. Potremmo anche migliorare la grafica, introdurre il colore, il suono (quando si risponde giusto si sentirà la nota, per esempio). Potremmo anche realizzare seguendo lo stesso principio un gioco che vi permetta di riconoscere degli accordi, ecc.

```
10 REM SOLFEGGIO
20 DIM R$(3)
30 S=0:P=0
40 PRINT "}"
50 FOR I=1 TO 5
60 POSITION 1,2*I+1:PRINT "_____________________________________"
70 NEXT I
80 X=3+INT(RND(1)*12)
90 POSITION 20,X-3:PRINT ""
100 POSITION 20,X-2:PRINT ""
110 POSITION 18,X-1:PRINT ""
120 POSITION 18,X:PRINT ""
130 POSITION 9,20:PRINT "PUNTI ";S;" IN ";P;" PROVE"
140 POSITION 2,22:PRINT " CHIAVE DI SOL - QUAL E' LA NOTA ";:INPUT R$
150 IF R$="" THEN 140
160 IF R$="0" THEN END
170 Y=2+1*R$="SOL"+2*R$="FA"+3*R$="MI"+4*R$="RE"+5*R$="DO"
180 Y=Y+6*R$="SI"+7*R$="LA"
190 P=P+1
200 IF Y=X OR (Y=X-7 AND X>9) THEN S=S+1
210 GOTO 40
```

Blackjack

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: contro l'ATARI.
Difficoltà del gioco: bassa.

IL GIOCO: Bisogna inserire una scommessa all'inizio del gioco; e potremo poi lanciare i dadi battendo 1; ad ogni lancio si otterrà un nuovo totale, che aumenterà poco a poco e si avvicinerà a 21; bisognerà cercare di avvicinarsi a 21 senza superarlo. Quando decideremo di fermarci, basterà battere 0 per lasciar giocare l'ATARI che cercherà allora di ottenere un miglior punteggio.
Alla fine di ogni partita sarà visualizzato la vincita così come il totale delle vincite (o delle perdite).

IL PROGRAMMA: Le linee 80 e 90 realizzano il controllo di quello che deve giocare (giocatore o ATARI).
Il sottoprogramma (1000-1010) lancia i dadi.
Le linee 110 e 120 calcolano il totale fatto dal giocatore e lo visualizzano.
Il totale dell'ATARI è calcolato alla linea 190 e visualizzato alla linea 200.
La linea 220 controlla la fine del gioco per l'ATARI.
Le linee 250-280 visualizzano i risultati e la linea 290 cerca qual è il vincitore.
La vincita (o la perdita) sarà visualizzata dalla linea 310 ed il totale dalla linea 340.
La linea 370 controlla la fine della partita.

PER GIOCARE: Battere RUN per iniziare il gioco.
Alla domanda LA TUA PUNTATA, inserire la scommessa iniziale, seguita da RETURN, e battere 1 per lanciare il dado. Il primo tiro è visualizzato sotto la forma IL TUO TOTALE RAGGIUNGE . . .; si dovrà ribattere 1 seguito da RETURN, finché si desidera ricominciare a tirare.
Appena finito, si battono 0 e RETURN per lasciar giocare l'ATARI (che ha qui il ruolo di banco).
La vincita (o la perdita, se c'è il segno —) apparirà alla fine del gioco, così come il totale delle vincite (o delle perdite) realizzate dall'inizio.
Se si desidera interrompere il gioco, si dovrà battere N in risposta alla domanda CONTINUIAMO? (S/N).

ESTENSIONI POSSIBILI: Questo gioco sarà evidentemente più interessante se è fatto in tanti, e sarà facile modificare il programma di conseguenza; potremmo anche disegnare il dado , o meglio, utilizzare delle carte, il che permetterà allora di fare il vero Blackjack.

```
10 REM BLACKJACK
20 DIM R$(1)
30 G=0
40 GRAPHICS 0
50 T=0
60 PRINT "     LA TUA PUNTATA :";:INPUT M
70 PRINT
80 PRINT "Battere 1 per tirare - 0 per finire";:INPUT R
90 IF R=0 THEN 140
100 GOSUB 1000
110 T=T+X
120 PRINT "Il tuo totale raggiunge ";T
130 GOTO 70
140 PRINT "}"
150 PRINT "          IL BANCO GIOCA"
160 PRINT
170 S=0
180 GOSUB 1000
190 S=S+X
200 PRINT "Il banco raggiunge ";S
210 FOR I=1 TO 500:NEXT I
220 IF S>T OR S>17 THEN 240
230 GOTO 180
240 PRINT "}"
250 PRINT "Il tuo totale e' di ";T
260 PRINT
270 PRINT "Quello del banco e' di ";S
280 PRINT
290 IF S>21 OR (S<T AND T<=21) THEN 310
300 M=-M
310 PRINT "La tua vincita e' di ";M
320 PRINT
330 G=G+M
340 PRINT "La tua vincita totale arriva a ";G
350 PRINT
360 PRINT "Continuiamo (S/N) ";:INPUT R$
370 IF R$="N" THEN END
380 GOTO 40
1000 X=INT(RND(1)*6+1)
1010 RETURN
```

Pari - dispari

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: media.

IL GIOCO: È un gioco d'azzardo per uno o più giocatori. L'ATARI lancia simultaneamente due dadi non visibili e fa tre domande: qual è la parità del totale, qual è il totale ed infine quale sia il valore di uno dei dadi.
Si può proseguire il gioco o interromperlo in ogni momento.

IL PROGRAMMA: I dadi sono tirati da un sottoprogramma posizionato alle linee 1000 e 1010.
La linea 70 tiene il conto del numero di partite e le linee 170, 250 e 280 contabilizzano i punti segnati.
La parità del totale è determinata dalle linee 120 e 130 (la linea 130 paragona con la risposta del giocatore).
La linea 400 e le seguenti realizzano il messaggio di fine partita: totale, numero di punti, numero di partite giocate.
La linea 440 provoca l'arresto se si risponde N alla domanda CONTINUIAMO? (S/N)

PER GIOCARE: Si lancia il programma con RUN.
Basta in seguito rispondere alle diverse domande poste, seguendo scrupolosamente le indicazioni date e battendo RETURN dopo ogni risposta.
Quando si finisce una partita, si ottiene la visualizzazione del totale attuale ed il numero di partite nella forma di, per esempio, 6 punti in 10 prove.
Si ottiene allora la domanda CONTINUIAMO? (S/N) e si risponderà N seguito da RETURN per fermarsi.

ESTENSIONI POSSIBILI: Se si desidera giocare in tanti, sarebbe interessante predisporre la possibilità di scommettere e di aggiungere una gestione del punteggio per ciascuno dei giocatori. Potremmo anche prevedere una miglioria nella presentazione (utilizzo del colore, ecc.)

```
10 REM PARI-DISPARI
20 DIM R$(1)
30 T=0:R=0
40 GOSUB 1000
50 A=X
60 GOSUB 1000
70 R=R+1
80 S=A+X
90 PRINT "}"
100 PRINT "Totale pari (0) o dispari (1) :";:INPUT N
110 PRINT
120 P=S/2-INT(S/2)
130 IF N=P OR N-0.5=P THEN 160
140 PRINT "SBAGLIATO!"
150 GOTO 180
160 PRINT "ESATTO."
170 T=T+1
180 PRINT
190 PRINT "Qual e' il totale presunto:";:INPUT N
200 IF N=S THEN 240
210 PRINT :PRINT "PERSO!"
220 FOR I=1 TO 900:NEXT I
230 GOTO 400
240 PRINT
250 T=T+2
260 PRINT "BRAVO! Fornisci uno dei valori :";:INPUT N
270 IF N<>A AND N<>S-A THEN 210
280 T=T+3
290 PRINT
300 PRINT "BRAVO!":GOTO 220
400 PRINT "}"
410 PRINT "Hai ";T;" punti in ";R;" prove."
420 PRINT
430 PRINT "Continuiamo (S/N) ";:INPUT R$
440 IF R$="N" THEN END
450 GOTO 40
1000 X=INT(RND(1)*6+1)
1010 RETURN
```

Indovina un numero

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: facile.

IL GIOCO: È un gioco d'azzardo molto semplice, ma molto interessante: il giocatore introduce la sua puntata e scommette su un numero, da 1 a 6; sono allora lanciati tre dadi; se il numero giocato appare due volte raddoppia la sua puntata; se appare tre volte, la triplica.

IL PROGRAMMA: Il lancio dei dadi è realizzato da un sottoprogramma posizionato alle linee 1000 e 1010.
La visualizzazione è realizzata dalle linee 140 e 150.
Le linee da 170 a 200 calcolano il guadagno nelle diverse configurazioni possibili.
La totalizzazione è effettuata dalla linea 220 per le vincite e dalla linea 510 per le perdite.
La variabile R$, verificata alla linea 550, determina il proseguimento eventuale del gioco.

PER GIOCARE: Il lancio del programma con RUN fa apparire un primo messaggio LA TUA PUNTATA; si risponde con un numero seguito da RETURN. Si ottiene allora un secondo messaggio FORNISCI UN NUMERO DA 1 A 6; si inserisce un numero seguito da RETURN; i tre dadi, raffigurati dalle lettere A, B, C appaiono allora, assieme a un messaggio indicante la vincita o la perdita.
Poi appare un'indicazione sul vostro totale attuale appare, con la domanda CONTINUIAMO (S/N)?; battere S per continuare, N per fermarsi (seguito da RETURN).

ESTENSIONI POSSIBILI: Potremmo facilmente introdurre dei controlli (puntata minima, colpo proposto da 1 a 6, ecc.). Potremmo anche disegnare dei dadi con una grafica più elaborata, introdurre la possibilità di giocare in molti, o anche fare partecipare l'ATARI.

```
10 REM INDOVINA UN NUMERO
20 DIM R$(1)
30 G=0
40 PRINT "}"
50 PRINT "La tua puntata :";:INPUT N
60 PRINT
70 PRINT "Fornisci un numero da 1 a 6 :";:INPUT D
80 GOSUB 1000
90 A=X
100 GOSUB 1000
110 B=X
120 GOSUB 1000
130 C=X
140 PRINT :PRINT :PRINT "    A          B            C"
150 PRINT :PRINT "    ";A;"          ";B;"            ";C
160 T=0
170 IF D=A THEN T=T+N
180 IF D=B THEN T=T+N
190 IF D=C THEN T=T+N
200 IF T=0 THEN 500
210 PRINT :PRINT :PRINT "Vinci ";T;"."
220 G=G+T
230 GOTO 520
500 PRINT :PRINT :PRINT "PERSO!          "
510 G=G-N
520 PRINT :PRINT :PRINT "Totale vincite :";G;"."
530 POSITION 9,23
540 PRINT "Continuiamo (s/N) ";:INPUT R$
550 IF R$="N" THEN END
560 GOTO 40
1000 X=INT(RND(1)*6+1)
1010 RETURN
```

Il gioco di Wytthof

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO: Disponiamo di una griglia quadrata di 10 * 10 caselle. Un punto appare, al di fuori della diagonale e i giocatori lo dovranno spostare a turno, seguendo solamente tre direzioni: destra, diagonale in basso a destra, basso. Il primo giocatore che arriva alla casella (9,9) sarà considerato come vincente, o perdente, secondo la convenzione presa all'inizio.

IL PROGRAMMA: Le linee 30, 40 e 50 permettono di definire una posizione di partenza, al di fuori della diagonale.
La linea 70 e il ciclo I (80-100) disegnano la griglia.
La linea 120 stampa la posizione del punto mobile.
Il test di fine è effettuato alla linea 130.
Il cambiamento del giocatore è assicurato dalla linea 170.
Le direzioni di spostamento disponibili, 1, 2 e 3, sono decodificate da tre sottoprogrammi che cominciano alle linee 1000, 2000 e 3000.

PER GIOCARE: Battere RUN per lanciare l'esecuzione del programma. Il campo di gioco appare, così come la posizione di partenza.
Poi si ottiene il messaggio GIOCATORE 1 e sotto la domanda DIREZIONE (1,2,3), alla quale si risponde battendo uno di questi tre numeri, seguito da RETURN.
Sarà in seguito il turno del giocatore 2.
Il gioco termina quando uno dei giocatori arriva alla casella (9,9).

ESTENSIONI POSSIBILI: Possiamo innanzitutto migliorare i controlli (l'uscita dal terreno non è proibita, per esempio), poi la visualizzazione (grafica, colori), estendere il gioco a vari giocatori con un campo più grande, introdurre un mostro che cercherà di catturare uno dei giocatori, predisporre un programma che permetta all'ATARI di giocare, ecc.

```
10 REM IL GIOCO DI WYTTHOF
20 DIM D$(1)
30 X=3*INT(RND(1)*5)+5
40 Y=INT(RND(1)*5)
50 IF 3*Y+5=X THEN 50
60 GRAPHICS 0
70 PRINT "   0  1  2  3  4  5  6  7  8  9"
80 FOR I=0 TO 9
90 PRINT :PRINT I;"  .  .  .  .  .  .  .  .  .  ."
100 NEXT I
110 H=1
120 POSITION X,2*Y+2:PRINT ""
130 IF X=32 AND Y=9 THEN END
140 POSITION 4,22:PRINT "Giocatore ";H;:PRINT " - Direzione (1,2,3) ";:INPUT D$
150 D=VAL(D$)
160 ON D GOSUB 1000,2000,3000
170 H=2-ABS(H-1)
180 GOTO 120
1000 X=X+3
1010 RETURN
2000 GOSUB 1000
3000 Y=Y+1
3010 RETURN
```

Rimbalzi

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di abilità.
Difficoltà del gioco: media.

IL GIOCO: Utilizza l'idea della simulazione di un biliardo.
Alle due estremità dello schermo si trovano la palla ed il bersaglio: una serie di cifre e di lettere, in alto sullo schermo, permettono di scegliere il posto dove si desidera rimbalzare.
La palla partirà allora verso l'alto (verso il basso se si introduce il segno —), rimbalzerà, ritornerà eventualmente in basso, rimbalzerà di nuovo, e così via fino a raggiungere la linea del bersaglio; se si è sufficientemente vicini al bersaglio si vince ed il numero di rimbalzi sarà allora visualizzato.

IL PROGRAMMA: La linea 40 stampa le cifre e le lettere in alto sullo schermo.
Le linee 50 e 60 definiscono le posizioni iniziali della palla e del bersaglio.
Le linee 70 e 80 effettuano la loro stampa.
Il simbolo corrispondente alla direzione scelta è inserito alla linea 90.
Il ciclo J (120-160) realizza lo spostamento della palla.
La linea 150 calcola la nuova posizione.
Il cambiamento di direzione, durante lo scontro con un bordo, è effettuato dalla linea 140, per rinvio in una parte indipendente del programma, situata alle linee da 210 a 230.
La linea 170 effettua la stampa della fine della partita.
Il numero di rimbalzi è contato dalla linea 220.

PER GIOCARE: Si lancia il programma con RUN.
Il campo appare, così come le posizioni iniziali della palla e del bersaglio. Si sceglie il posto dove si desidera far rimbalzare la palla (da 1 a Z), e lo si inserisce.
La traiettoria è allora mostrata, seguita, se lo si merita, dal messaggio BRAVO; X RIMBALZI.
Si può cominciare un'altra partita battendo un tasto qualunque, e fermarsi in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Sarà molto interessante, utilizzando una grafica più fine, disegnare un vero biliardo, col suo tappeto verde, ed indicare con un segmento, che simboleggia la stecca, che si sposta intorno alla biglia.
Potremmo anche aggiungere delle buche, mettere varie biglie, sonorizzare i rimbalzi, o gli urti con altre palle, contabilizzarle per fare una partita con vari giocatori, ecc.

```
10 REM RIMBALZI
20 DIM R$(1)
30 GRAPHICS 0
40 POSITION 0,0:PRINT "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
50 X=INT(RND(1)*18+2)
60 Y=INT(RND(1)*18+2)
70 POSITION 0,X:PRINT CHR$(1)
80 POSITION 39,Y:PRINT CHR$(4)
90 POSITION 0,22:PRINT "Il tuo colpo : ";:INPUT R$
100 S=0
110 T=X
120 FOR J=1 TO 38
130 T=T-X/(ASC(R$)-ASC("0")-7*(R$>="A"))
140 IF T>21 OR T<1 THEN 210
145 IF J=39 THEN 160
150 POSITION J,T:PRINT CHR$(20)
160 NEXT J
170 IF ABS(T-Y)<2 THEN POSITION 0,21:PRINT "BRAVO! ";S;" rimbalzi.";
180 POSITION 0,22:PRINT "Un'altra partita (S/N) ";:INPUT R$
190 IF R$="N" THEN END
200 RUN
210 X=-X
220 S=S+1
230 GOTO 160
```

Jackpot

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: fortuna (un giocatore).
Difficoltà del gioco: facile.

IL GIOCO: È una variante semplificata di un gioco d'azzardo classico, nel quale il giocatore deve fare una scommessa. Se ottiene certe combinazioni di simboli, vince (più o meno, secondo la sua puntata e la combinazione ottenuta).
Qui, i simboli sono sostituiti da cifre, e il fatto di avere tre cifre identiche permette di vincere. Esiste più di una combinazione particolare, che permette di avere il Jackpot.

IL PROGRAMMA: Il ciclo I (60-80) permette il disegno della macchina, la linea 90 visualizza gli spostamenti (vuoti) delle cifre.
La scommessa sarà inserita alla linea 100, e testata alla linea 110: se si inserisce 0 la partita è terminata (partenza verso la linea 2000).
Le linee 120, 140 e 160 permettono il lancio delle tre cifre, richiamando un sottoprogramma che si trova alla linea 1000.
Il risultato è visualizzato alla linea 170, poi testato alla linea 180 e alla linea 310, se si vince.

PER GIOCARE: Lanciare il programma con RUN; alla domanda CAPITALE INIZIALE?, si risponde inserendo un numero sufficientemente alto, dal quale saranno dedotte le vostre puntate.
Poi si batte RETURN, e si inserisce una prima puntata (differente da 0), seguita da RETURN.
Le tre cifre estratte sono allora visualizzate un breve istante, così come la puntata fatta, ed il risultato ottenuto (vincita o perdita).
Si può in seguito scommettere di nuovo, fare RETURN e così via.
Per terminare, basta rispondere 0 alla domanda LA TUA PUNTATA?
Apparirà allora il resto del vostro capitale iniziale.

ESTENSIONI POSSIBILI: Potremmo migliorare seriamente i controlli, usare i colori e delle possibilità grafiche più elaborate per simulare meglio un vero Jackpot.

```
10 REM JACKPOT
20 DIM R$(1)
30 PRINT "Capitale iniziale : ";:INPUT P
40 GRAPHICS 0
50 SETCOLOR 1,0,0
60 FOR I=1 TO 3
70 POSITION 10,I+4:FOR N=1 TO 21:PRINT CHR$(160);:NEXT N
80 NEXT I
90 POSITION 14,6:PRINT "   ":POSITION 19,6:PRINT "   ":POSITION 24,6:PRINT "   "
100 POSITION 12,11:PRINT "La tua puntata :";:INPUT M
110 IF M=0 THEN 2000
120 GOSUB 1000
130 A=X
140 GOSUB 1000
150 B=X
160 GOSUB 1000
170 POSITION 15,6:PRINT A:POSITION 20,6:PRINT B:POSITION 25,6:PRINT X
180 IF A=B AND A=X THEN 300
190 N=-M
200 POSITION 8,11:PRINT "Puntata :";M;" - Risultato :";N;"     "
210 POSITION 4,22:PRINT "Battere <RETURN> per continuare ";:INPUT R$
220 P=P+N
230 GOTO 40
300 N=M*50
310 IF A=7 THEN N=M*500
320 GOTO 200
1000 X=INT(RND(1)*9+1)
1010 RETURN
2000 PRINT "}"
2010 PRINT "Hai ";P;
2020 PRINT " punti.":PRINT "A presto!"
```

Roulette

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: fortuna (vari giocatori).
Difficoltà del gioco: facile.

IL GIOCO: È soprattutto spettacolare per il suo aspetto grafico: una pallina, raffigurata da un quadrato nero, si sposta su un campo di gioco composto da caselle numerate da 1 a 36, così come le tre caselle supplementari, la 00, una casella P (come pari) e una casella D (come dispari).
La pallina percorre almeno una volta tutte le caselle, poi si ferma al secondo giro; si può in seguito rilanciarla per un'altra partita.

IL PROGRAMMA: Il ciclo K (50-100) disegna il tavolo di gioco, al quale la linea 110 aggiunge i tre simboli P, 00, e D.
Il ciclo K (120-230) realizza lo spostamento del ciclo, l'arresto è controllato dalla linea 160 e 210.
Il sottoprogramma 300-310 realizza una leggera attesa.
La linea 240 sceglie la posizione di arresto.
La linea 260 gestisce i tentativi seguenti.
È da notare che la pallina è il cursore che si sposta.

PER GIOCARE: Basta battere RUN, e guardare.
Una volta che la pallina si sia fermata basterà battere un tasto qualunque (tranne "N") per rilanciare.
Battere N per uscire dal programma.

ESTENSIONI POSSIBILI: Potremmo cercare di avvicinarci maggiormente ad una vera roulette, in diversi modi: disegnare meglio il tavolo, introdurre i colori, simulare il rallentamento della pallina, introdurre differenti possibilità di scommessa, e assicurare la gestione delle vincite corrispondenti, tenere i conti di vari giocatori, ed annunciare i risultati in una maniera più usuale.

```
10 REM ROULETTE
20 DIM R$(1)
30 GRAPHICS 0
40 SETCOLOR 2,12,2
50 FOR K=1 TO 4
60 FOR I=1 TO 9
70 S=2*I
80 POSITION 4*I,4*K:PRINT S*(K=1)+(S+18)*(K=2)+(S-1)*(K=3)+(S+17)*(K=4)
90 NEXT I
100 NEXT K
110 POSITION 1,6:PRINT "P":POSITION 1,10:PRINT "00":POSITION 1,14:PRINT "D"
120 FOR K=1 TO 4
130 FOR I=1 TO 9+(K<4)
140 IF I=10 AND K<4 THEN POSITION 0,4*K+3:PRINT " ":GOTO 160
150 POSITION 4*I-1,4*K+1:PRINT " ";
160 X=X-1
170 SOUND 0,53,10,4
180 FOR T=1 TO 30:NEXT T
190 SOUND 0,0,0,0
200 GOSUB 300
210 IF X=1 THEN PRINT CHR$(20):GOTO 260
220 NEXT I
230 NEXT K
240 X=INT(RND(1)*38)
250 GOTO 110
260 POSITION 0,20:PRINT "Un'altra prova (S/N) ";:INPUT R$
270 IF R$="N" THEN END
280 GOTO 40
300 FOR J=1 TO 100:NEXT J
310 RETURN
```

Corridoi

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessione.
Difficoltà del gioco: elevata.

IL GIOCO: Fa richiamo a delle nozioni di logica elementare. Il giocatore può presentarsi all'ingresso sia del corridoio n. 1 che del n. 2. In questi due ingressi sono piazzati degli sportelli che si spostano al vostro passaggio e vi lasciano passare, una volta a destra, una volta a sinistra. In più uno sportello analogo è stato messo nel corridoio centrale; potrete prevedere dove (uscita A o B) si effettuerà la vostra prossima uscita?

IL PROGRAMMA: Le linee da 40 a 110 disegnano i corridoi (e i tre sportelli).
La linea 140 aspetta la vostra scelta .
La decodifica della strada è fatta dall'ATARI utilizzando tre variabili logiche I,J,K, che rappresentano lo stato dei tre sportelli (linee da 180 a 320).

PER GIOCARE: Dopo aver battuto RUN, il corridoio è disegnato sullo schermo, gli sportelli sono visibili e bisognerà battere 1 o 2 in risposta alla domanda dell'ATARI.
Nella versione presentata allora immediatamente visualizzate la risposta e la strada seguita dall'inizio del gioco; anche le uscite corrispondenti appaiono sullo schermo. Ciò presenta il vantaggio di permettere a vari giocatori di esercitare la loro sagacità cercando di prevedere, per un certo ingresso, ciò che risponderà l'ATARI.

ESTENSIONI POSSIBILI: Sono molto numerose; è molto facile immaginare un corridoio più lungo e complesso, con altri ingressi ed altre uscite, e altri sportelli. Possiamo immaginare una visualizzazione grafica dei risultati disegnando, per un breve istante, lo stato degli sportelli dopo la risposta del giocatore. Possiamo anche aggiungere un test delle risposte (gioco in tanti), contabilizzare i punti di ogni giocatore, ecc.
Nota per il programma: l'ATARI dispone di caratteri grafici disponibili battendo contemporaneamente CONTROL e uno dei tasti alfabetici.

```
10 REM CORRIDOI
20 DIM C$(1),R$(37),S$(1),T$(37)
30 GRAPHICS 0
40 PRINT "nnnnnnnnnnnnnnnnnnnnnnn":REM per ogni lettera minuscola, battere simult
aneamente il tasto <CONTROL>
50 PRINT :PRINT "1  gf  nnnnnnnnnnn     A":REM vedere 40
60 PRINT "   fg":REM vedere 40
70 PRINT "nnnnnnnn      gf  nnnnnn":REM vedere 40
80 PRINT "              fg":REM vedere 40
90 PRINT "2  gf  nnnnnnnnnnn     B":REM vedere 40
100 PRINT "   fg":REM vedere 40
110 PRINT "nnnnnnnnnnnnnnnnnnnnnnn":REM vedere 40
120 I=0:J=0:K=0
130 R$="":T$=""
140 POSITION 0,13:PRINT "Entrata 1 o 2 :       ":POSITION 17,13:INPUT C$
150 IF C$="" OR C$>="3" OR LEN(C$)<>1 THEN 140
160 R$(LEN(R$)+1)=C$
170 PRINT :PRINT R$
180 IF C$="1" THEN 230
190 J= NOT (J)
200 IF J<>0 THEN 300
210 S$="B"
220 GOTO 260
230 I= NOT (I)
240 IF I<>0 THEN 300
250 S$="A"
260 T$(LEN(T$)+1)=S$
270 POSITION 0,17:PRINT "Uscito in ";S$:PRINT :PRINT T$
280 IF LEN(T$)>36 THEN RUN
290 GOTO 140
300 K= NOT (K)
310 IF K<>0 THEN 210
320 GOTO 250
```

Scacchiera

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: grafica.
Difficoltà del gioco: secondo l'utilizzo.

IL GIOCO: Il programma proposto viene in appoggio al programma Scacchi; esso dà, infatti, la possibilità di realizzare una visualizzazione grafica di buona qualità per i giochi che utilizzano una scacchiera di 8 caselle (potremo facilmente estenderla a 10 * 10). È stata prevista una locazione: orizzontalmente con delle lettere (da A ad H) e verticalmente con dei numeri (da 1 a 8).

IL PROGRAMMA: Le linee 50 e 60 definiscono due variabili A$ e B$ che contengono gli elementi del disegno. La linea 70 stampa le lettere orizzontali.
Il ciclo K (120-160) disegna una casella.
Il ciclo J (110-170) disegna una linea.
Il ciclo I (90-180) disegna le linee con il loro numero.

PER GIOCARE: L'ordine RUN provoca l'apparizione della scacchiera sullo schermo.
Per utilizzare il programma, converrà tener conto della localizzazione delle caselle utilizzabili che sono i centri dei quadrati. Le caselle sono ottenute in larghezza da una tabulazione del tipo 3 * J,J variante da 1 a 8, e in altezza da 2 * I+K

ESTENSIONI POSSIBILI: Potremmo usare questo programma per tutta una serie di giochi che si svolgano su una scacchiera (scacchi, dama, il lupo e gli agnelli, ecc.).

```
10 REM SCACCHIERA
20 PRINT "}"
30 SETCOLOR 1,9,14:SETCOLOR 2,0,0
40 DIM A$(3),B$(3)
50 A$="   "
60 B$="   ":REM battere questi 3 bianchi in caratteri inversi
70 PRINT "   A  B  C  D  E  F  G  H"
80 PRINT
90 FOR I=1 TO 8
100 PRINT I
110 FOR J=1 TO 8
120 FOR K=1 TO 2
130 POSITION 3*J+1,2*I+K
140 IF INT((I+J)/2)=(I+J)/2 THEN PRINT A$:GOTO 160
150 PRINT B$
160 NEXT K
170 NEXT J
180 NEXT I
```

Il gioco della fava

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO: Presenta una certa analogia con il gioco di WYTT-HOF; questa versione offre però 4 possibilità di spostamento; la posizione di partenza è qui fissata in (0,0) e potrebbe eventualmente essere modificata.
Le direzioni di spostamento disponibili sono: destra, basso, diagonale in basso a destra, diagonale in basso a sinistra.
Si gioca con la regola: il primo arrivato in (9,9) ha perso.

IL PROGRAMMA: La linea 40 ed il ciclo I (50-70) disegnano il campo di gioco.
La linea 110 realizza il test di fine partita.
La linea 140 permette di eseguire uno dei sottoprogrammi di spostamento.
Le linee 150 e 160 assicurano il cambiamento di giocatore.

PER GIOCARE: Si lancia il programma con RUN ed appare il campo di gioco , così come la domanda DIREZIONE (1, 2, 3, 4); le 4 direzioni proposte corrispondono agli spostamenti seguenti:

Il giocatore 1 inserisce il suo spostamento e batte RETURN.
Il movimento corrispondente è realizzato, poi si ottiene il messaggio GIOCATORE 2 e DIREZIONE.
Il giocatore 2 procede nello stesso modo e così via, fino a che uno dei giocatori arriva in (9,9) nel qual caso il gioco termina.

ESTENSIONI POSSIBILI: Si può immaginare di introdurre all'inizio della partita delle caselle con delle trappole (ognuno cercherà allora di fare finire l'altro in una di quelle caselle).
Si può anche giocare con vari simboli, se possibile di colore diverso, spostati simultaneamente, proporre dei campi di gioco triangolari, di dimensione arbitraria, ecc.

```
10 REM IL GIOCO DELLA FAVA
20 GRAPHICS 0
30 DIM I$(1)
40 PRINT "   0  1  2  3  4  5  6  7  8  9"
50 FOR I=1 TO 9
60 PRINT :PRINT I;"  .  .  .  .  .  .  .  .  .  ."
70 NEXT I
80 X=5:Y=1
90 J=1
100 POSITION X,2*Y:PRINT CHR$(20)
110 IF X=32 AND Y=9 THEN 200
120 POSITION 5,21:PRINT "Giocatore ";J;" - Direzione(1,2,3,4)";:INPUT I$
130 I=VAL(I$)
140 ON I GOSUB 1000,2000,3000,4000
150 J=J+1
160 IF J=3 THEN 90
170 GOTO 100
200 POSITION 0,21:PRINT "Il giocatore ";J;" vince.....                ";
210 END
1000 Y=Y+1
1010 RETURN
2000 GOSUB 1000
3000 X=X+3
3010 RETURN
4000 GOSUB 3000
4010 Y=Y-1
4020 RETURN
```

Gioco dell'oca casuale

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: fortuna (due giocatori).
Difficoltà del gioco: bassa.

IL GIOCO: Il gioco riprende l'antichissima idea del gioco dell'oca, ma aggiungendo la possibilità offerta dal calcolatore di scegliere a caso le caselle "con problema".
La regola del gioco non ha molto interesse, visto che i giocatori non intervengono in nessun momento nello svolgimento della partita, che è interamente gestita dall'ATARI; basta schiacciare uno dei tasti dopo il lancio del dado e guardare cosa succede.

IL PROGRAMMA: La linea 60 permette di chiamare il sottoprogramma 1000-1100 che visualizza il tavolo da gioco, il controllo della posizione dei giocatori e la visualizzazione di quest'ultima. Questa visualizzazione sarà d'altronde rifatta durante la partita, per tener conto dello spostamento realmente effettuato (che non è necessariamente quello indicato dal dado).
La linea 60 si occupa del cambiamento del giocatore e la linea 70 effettua il lancio del dado.
La linea 80 si occupa del caso in cui uno dei giocatori termina.
La linea 120 prende un numero casuale tra 0 e 3, che servirà, se sarà il caso, a determinare il vero spostamento del giocatore, rinviando in uno dei sottoprogrammi di "fortuna".
Le linee 1000 e 1010 fanno avanzare le pedine e le linee 1020-1080 ridisegnano il tavolo da gioco ad ogni turno.

PER GIOCARE: Si lancia il gioco con RUN. Il tavolo da gioco è allora disegnato, poi appare il messaggio GIOCATORE 1: . . . LA TUA MOSSA
Si dovrà allora battere su un tasto qualsiasi perché il colpo del giocatore 1 sia preso in considerazione; non preoccupatevi per il tempo di reazione, che potrà sembrare un pò lungo.
Sarà eseguito lo spostamento indicato dai dadi poi, dopo qualche istante, sarà fatto eventualmente un secondo spostamento: è lo spostamento "FORTUNA".
Apparirà allora il messaggio GIOCATORE 2: . . .
Il giocatore 2 batte a sua volta un tasto qualsiasi e così via...
Il giocatore vincente sarà segnalato da un messaggio BRAVO in fine di partita.
Si può fermare il gioco in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Si potrebbe ingrandire il tavolo da gioco, introdurre un maggior numero di messaggi casuali, cambiare la loro natura. Potremmo anche scegliere le caselle "fortuna" all'inizio del gioco, aggiungere dei messaggi di conseguenza, ecc.

```
10 REM GIOCO DELL'OCA CASUALE
20 GRAPHICS 0
30 DIM R$(1)
40 X=0:Y=0:J=1:I=X
50 GOSUB 1000
60 J=ABS(J-1)
70 I=INT(RND(1)*6+1)
80 IF X=35 OR Y=35 THEN POSITION 0,20:PRINT "BRAVO!":END
90 POSITION 6,12:PRINT "Giocatore ";J+1;" - il tuo colpo :";I
100 PRINT "     Battere <RETURN>";:INPUT R$
110 GOSUB 1000
120 I=INT(RND(1)*4)
130 F=INT(RND(1)*5+1)
140 ON F GOSUB 200,300,400,500,150
150 GOTO 50
200 I=I+3
210 RETURN
300 I=-I
310 RETURN
400 I=2*I
410 RETURN
500 IF J=0 THEN X=INT(RND(1)*30):RETURN
510 Y=INT(RND(1)*30):RETURN
1000 IF J=0 THEN X=X+I*(X+I<=35)
1010 Y=Y+J*I*(Y+I<=35)
1020 POSITION 2,3:PRINT "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"
1030 FOR I=0 TO 35
1040 POSITION 2+I,5:PRINT CHR$(151)
1050 IF I=X THEN POSITION 2+I,5:PRINT "1";
1060 POSITION 2+I,6:PRINT CHR$(152)
1070 IF I=Y THEN POSITION 2+I,6:PRINT "2";
1080 NEXT I
1090 FOR I=1 TO 400:NEXT I
1100 RETURN
```

LIVELLO 2

N.	Nome del programma	Tema e particolarità
1	Dado truccato	Da utilizzare, se siete disonesti, in altri programmi
2	Visioni	Un gioco sulle parole, per il calcolatore
3	421	Una simulazione semplificata per vari giocatori
4	L'elettricista pazzo	Logico e difficile, per 1 solo giocatore
5	La corsa al 20	Gioco di riflessione, per 2 giocatori o più
6	Musica	Ripassate le vostre nozioni di solfeggio, con l'aiuto del calcolatore
7	Cavallina	Un gioco di riflessione, per 2 giocatori
8	Lettere	Un gioco sulle parole, abbastanza semplice
9	Macchina delle parole	Lasciate creare al calcolatore dei vocaboli nuovi
10	Spia	Ditegli il vostro amore in codice
11	Attaccate i vagoni	Per i più giovani; molto visivo
12	Lancio di un dado truccato	Ripassate le vostre conoscenze di statistica
13	Gioco con le carte	Il vostro calcolatore ha più di un trucco nel sacco
14	Il gioco di Marienbad	Un gioco di logica per 2 giocatori, in una versione estesa
15	L'indesiderato	Siete proprietario. Arriverete a mettere le mani su un inquilino indesiderato
16	Componi	Fare della musica, è prima di tutto scriverla...
17	Giorno della settimana	Approfittatene per fare il punto sui compleanni
18	Gioco della vita	Un grande classico, in due versioni: manuale e automatica

Dimensionamento di un array
Ricerca di un elemento in un array e in una stringa
L'istruzione DATA

L'ISTRUZIONE DIM, O ISTRUZIONE DI DIMENSIONAMENTO DI UN ARRAY

Abbiamo già incontrato l'istruzione DIM per limitare la lunghezza di una stringa di caratteri.

Le strutture degli array vengono correntemente utilizzate in programmazione, si rivelano anzi indispensabili in un gran numero di casi. Se volete, per esempio, inserire le vostre entrate e uscite per tutti i giorni del mese (supporremo, per semplificare, dei mesi di trenta giorni) potreste definire 30 variabili chiamate PRIMO GIORNO, SECONDO GIORNO, ecc. È più facile e più razionale utilizzare un ciclo I, I variante da 1 a 30, una variabile R(I) per le entrate e D(I) per le uscite; così facendo, avrete, senza saperlo, usato una struttura di array a una dimensione; questo sarà ammesso dall'ATARI solo se lo avete avvertito prima che avreste usato un tale array, dando un'istruzione DIM R(30) per le entrate e DIM D(30) per le uscite.

Potremmo anche immaginare di conservare le entrate e le uscite in uno stesso array, dandogli questa volta due dimensioni: per esempio DIM RD(30,2) (o DIM RD(2,30)). Così, con DIM RD(30,2) si converrà che RD(I,1) rappresenti le entrate e RD(I,2) le uscite. Sono ugualmente possibili degli array con più di due dimensioni.

In ATARI è obbligatorio dimensionare un array che contiene anche meno di 10 elementi. Attenzione alla creazione, gli elementi di un array non contengono obbligatoriamente un valore nullo.

LA RICERCA DI UN ELEMENTO IN UN ARRAY: R(I)

Se si ha un array a una dimensione, numerico o definito da un'istruzione come DIM R(30),l'Iesimo elemento di questo array è localizzato da R(I); sarà allora manipolato esattamente come una variabile ordinaria (assegnazione, calcolo, tests, ecc.).

Tuttavia non è possibile immettere un valore R(I) direttamente con il comando input.

LA RICERCA DI UN ELEMENTO IN UNA STRINGA: MID$

Per estrarre l'I-esimo elemento della stringa alfanumerica A$, si utilizzerà la forma A$(I,I).

Così, se A$="BUONGIORNO", A$(3,3) darà il carattere N; inoltre A$(3,6) darà la sottostringa "ONG".

IL TRATTAMENTO DI DATI MULTIPLI ISTRUZIONI READ, DATA, RESTORE

Per riempire facilmente un array numerico o alfanumerico, sarà interessante utilizzare delle linee DATA che contengono i dati iscritti gli uni al seguito degli altri separati da una virgola.

Così si potrà scrivere:

DATA LEONE, GIULIO, Via MARTIN...

Questi dati possono essere letti nei cicli FOR...NEXT, tramite l'istruzione READ. Si può ricominciare la lettura all'inizio dei DATA, con l'aiuto dell'istruzione RESTORE.

È possibile verificare una linea di DATA particolare con l'aiuto dell'istruzione RESTORE XXX dove XXX è il numero del passo del programma.

Dado truccato

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: media.

IL GIOCO: Si vuole cercare di fabbricare un dado truccato secondo la propria convenienza. Il metodo proposto qui consiste nel definire all'inizio una probabilità di realizzazione per ogni tipo di lancio (cioè la probabilità di ottenere un 1, un 2, un 3, ecc. che ognuna di queste probabilità sia compresa tra 0 e 1, e che la somma di tutte le probabilità sia un numero molto vicino a 1).

IL PROGRAMMA: Utilizza il generatore casuale dell'ATARI supposto di qualità adeguata (potremo facilmente usare un qualunque altro generatore se ce ne sarà bisogno). Il ciclo I (50-120) si occupa dell'ingresso di ognuna delle probabilità, con verifica ad ogni ingresso e sul totale (linea 90); ogni probabilità è conservata in una variabile indicizzata E(I): linea 70 del programma.
Il lancio del dado truccato è effettuato con l'aiuto di un altro ciclo I (160-180), la linea che permette il trucco è la 170. Un nuovo lancio potrà essere effettuato battendo un RETURN (linea 200).
Le linee dalla 300 alla 380 costituiscono un sottoprogramma di musica, estraendo valori di suoni dalla stringa A$.

PER GIOCARE: Dopo la partenza con la battuta di RUN, si ottiene la domanda PROBABILITÀ DEL DADO 1, alla quale si dovrà rispondere con un numero compreso tra 0 e 1. Converrà ricordarsi che il totale di tutte le probabilità inserite non dovrà in alcun caso superare 1.1 (tolleranza di 0.1). Dei messaggi di errore potranno apparire ed il programma sarà eventualmente rilanciato. Quando tutte le probabilità saranno state definite sarà automaticamente effettuato il primo lancio e basterà premere RETURN per effettuare i lanci successivi. Si arresta il programma battendo BREAK.

ESTENSIONI POSSIBILI: Ci sarebbe evidentemente la forte tentazione di utilizzare subdolamente questo piccolo programma per aumentare le vostre possibilità di vittoria in uno dei giochi d'azzardo che usano il lancio di dadi. Si potrebbe allora pensare di inserire i dati che definiscono il dado truccato in una variabile A$, ecc. Potremmo anche immaginare di presentare i risultati corrispondenti ad una serie di lanci in una maniera spettacolare, che metta il trucco in evidenza...

```
10 REM DADO TRUCCATO
15 DIM R$(1),VA$(9)
20 GRAPHICS 0
30 P=0
40 DIM E(6)
50 FOR I=1 TO 6
60 POSITION 0,3:PRINT "Probabilita' del dado ";I;
70 POSITION 27,3:INPUT E:E(I)=E
75 POSITION 0,3:PRINT "                                        "
80 P=P+E(I)
90 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 220
100 POSITION 6*I-3,6:PRINT "DADO ";I
110 POSITION 6*I-3,8:PRINT E(I)
120 NEXT I
130 IF P<0.9 THEN SOUND 0,32,10,4:GOSUB 400:SOUND 0,0,0,0:RUN
140 FOR I=1 TO 500:NEXT I
150 N=INT(RND(1)*6+1)
160 FOR I=1 TO 6
170 IF N=I AND RND(1)>E(I) THEN 150
180 NEXT I
190 POSITION 9,15:PRINT "Risultato del lancio :";N;" ";:GOSUB 300
200 PRINT :PRINT "Battere <RETURN> per lanciare il dado";:INPUT R$
210 GOTO 150
220 POSITION 0,4:PRINT "IMPOSSIBILE :0<=P<=1"
230 P=P-E(I)
240 FOR J=1 TO 1500:NEXT J
250 POSITION 0,4:PRINT "                                       "
260 GOTO 70
300 FOR I=0 TO 2
320 VA$="063040047":REM " DOSOL MI"
330 NO=VAL(VA$(1+3*I,3+3*I))
340 SOUND 0,NO,10,4
350 GOSUB 400
360 NEXT I
370 SOUND 0,0,0,0
380 RETURN
400 FOR T=1 TO 40:NEXT T
410 RETURN
```

Visioni

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: disegno.
Difficoltà del gioco: media.

IL GIOCO: L'ATARI vi permette, per la prima volta, di entrare nella sua intimità lasciandovi intravvedere la sua propria rappresentazione del mondo! Dovrete infatti proporgli delle parole, non importa quali, e lui vi risponderà proponendovi una raffigurazione grafica di ciò che rappresenta quella parola per lui. Il gioco sarà molto più divertente se lo farete in molti: potrete a volte avere delle sorprese.

IL PROGRAMMA: Le linee 50-80 scelgono in modo casuale 2 colori diversi per lo sfondo e le immagini.
Un ciclo I (90-170) si incarica di realizzare i disegni.
La linea 100 permette di trasformare in numero ogni lettera della parola inserita all'inizio (linea 40). Questo numero sarà in seguito portato in base 2 dal ciclo J (linee 110-150); secondo il risultato di questa trasformazione, un punto sarà stampato, oppure no (linea 140).

PER GIOCARE: Si inizia l'esecuzione del programma con RUN, il che porta all'apparizione della domanda NOME DEL DISEGNO? Si risponde inserendo una parola qualunque che sarà trasformata in disegno dall'ATARI non appena avremo battuto RETURN.
Per ricominciare basterà battere un tasto qualunque, per terminare battere BREAK.

ESTENSIONI POSSIBILI: La stessa idea può essere ripresa introducendo degli elementi di simmetria nei disegni, aggiungendo i colori. Possiamo anche immaginare di codificare con questo metodo un certo numero di disegni che introdurremo man mano nella macchina, e potremo allora creare un gioco nel quale l'ATARI diventerà sempre più esperto. Si può realizzare anche un gioco a due giocatori nel quale l'ATARI vi chiede di trovare la parola che corrisponde al disegno, ecc.

```
10 REM VISIONI
20 DIM R$(80)
30 GRAPHICS 0
40 PRINT "Nome del disegno :";:INPUT R$
50 TEXTE=INT(RND(1)*16):SETCOLOR 1,TEXTE,2*INT(RND(1)*8)
60 FOND=INT(RND(1)*16)
70 IF ABS(FOND-TEXTE)<1 THEN 60
80 SETCOLOR 2,FOND,6
90 FOR I=1 TO LEN(R$)
100 N=9*(ASC(R$(I,I)))-340
110 J=1
120 S=N/2-INT(N/2)
130 N=INT(N/2)
140 IF S<>0 THEN POSITION J+15,I+5:PRINT CHR$(160)
150 J=J+1
160 IF J<10 THEN 120
170 NEXT I
180 PRINT :PRINT
190 PRINT "Per continuare battere <RETURN>";:INPUT R$
200 GOTO 30
```

421

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: media.

IL GIOCO: Ogni giocatore dovrà, a turno, lanciare tre dadi e in seguito scegliere, nei due colpi successivi, se desidera o no rilanciare 1,2 o 3 dadi. Ottiene allora un punteggio corrispondente al numero indicato dai dadi, secondo le regole scelte, ed il giocatore successivo procede nella stessa maniera.

IL PROGRAMMA: Il numero di giocatori è innanzitutto inserito alla linea 30.
La linea 120 visualizza il numero di giocatori e il numero di tentativi già effettuati.
Un sottoprogramma posizionato alla linea 4000, utilizzato tre volte nel ciclo J (190-240), realizza il lancio dei dadi.
I rilanci sono eseguiti dalle linee 1000, 2000 e 3000.
La variabile R$ genera i secondi e terzi lanci. La stampa del risultato sarà in tutti i casi realizzata dalle linee 130-140.
I test per il numero di tentativi e per il cambio di giocatore sono effettuati alle linee 250 e 270.
Si osserverà la formula alla linea 180, molto utile per fare una concatenazione.

PER GIOCARE: Si lancia l'esecuzione del programma con RUN, il che porta al messaggio NUMERO DI GIOCATORI? Si inserisce questo numero, e si batte RETURN. Lo schermo si cancella e visualizza: GIOCATORE N. 1. PROVA A, PARTITA 1, così come il risultato realizzato, e il numero dei dadi (A, B, C).
Se non si desidera rifare dei lanci, basta battere RETURN. La mano passa allora al giocatore successivo.
Per effettuare un altro lancio, basta battere il nome del o dei dadi da rilanciare, e battere RETURN: per esempio, per ritirare A e C si batterà AC seguito da RETURN.
I giocatori successivi procederanno nello stesso modo.

ESTENSIONI POSSIBILI: La più evidente consiste nel lasciare la gestione dei risultati ai giocatori, riferendosi alle regole di vincita usuali del gioco 421. Potremmo fare apparire dopo ogni turno una tabella con la ricapitolazione dei risultati dei giocatori.
Potremmo anche migliorare la visualizzazione dei dadi (riferirsi ai vari programmi di visualizzazione dei dadi presenti nella stessa opera).

```
10 REM 421
20 DIM R$(6),X$(1)
30 PRINT "Numero di giocatori:";:INPUT N
40 P=1
50 I=1
60 K=0
70 PRINT "}"
80 R$="ABC"
90 GOTO 190
100 K=K+1
110 POSITION 17,1:PRINT "-";CHR$(180);CHR$(178);CHR$(177);"-"
120 POSITION 0,4:PRINT "Giocatore n. ";I;" - Prova ";K;" - Partita ";P
130 POSITION 8,7:PRINT A:POSITION 18,7:PRINT B:POSITION 28,7:PRINT C
140 POSITION 8,9:PRINT "A":POSITION 18,9:PRINT "B":POSITION 28,9:PRINT "C"
150 R$="":POSITION 0,14:PRINT "                                        "
160 POSITION 0,14:PRINT "Dado(i) da rilanciare (ABC) :       ";:POSITION 30,14:IN
PUT R$
170 IF R$="" THEN 260
180 R$(LEN(R$)+1)="DDD"
190 FOR J=1 TO 3
200 GOSUB 4000
210 X$=R$(J,J)
220 X=ASC(X$)-64
230 ON X GOSUB 1000,2000,3000,4000
240 NEXT J
250 IF K<3 THEN 100
260 I=I+1
270 IF I<=N THEN 60
280 P=P+1
290 GOTO 50
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=INT(RND(1)*6+1)
4010 RETURN
```

L'elettricista pazzo

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: elevata.

IL GIOCO: Un palazzo con 10 finestre, numerate da 0 a 9, appare sullo schermo. Il giocatore dovrà ad ogni tentativo manovrare uno degli interruttori, indicando il suo numero (da 0 a 9).
In teoria la finestra corrispondente dovrebbe spegnersi, ma è anche possibile che un'altra si illumini (o si spenga). Si tratta, naturalmente, di cercare di ottenere lo spegnimento delle 10 finestre col minor numero di tentativi possibile.

IL PROGRAMMA: Un array A, alla linea 30, conterrà lo stato (acceso o spento) delle 10 finestre.
Il ciclo I (60-80) e l'altro ciclo I (100-150) disegnano il palazzo nel suo stato iniziale.
La linea 160 controlla se tutto è spento, nel qual caso si andrà alla linea 300 per visualizzare il numero di tentativi.
La linea 170 aspetta l'indicazione dell'interruttore utilizzato.
Le linee da 180 a 280 definiscono il risultato della manovra.

PER GIOCARE: RUN provoca l'apparizione del palazzo con tutte le finestre accese.
Alla domanda INTERRUTTORE NUMERO?, si risponde indicando il numero dell'interruttore da usare, seguito da RETURN.
Il risultato della manovra è allora indicato sullo schermo (delle finestre si accendono o si spengono).
Si continua così fino a che tutte le finestre sono spente e si ottiene allora il numero di tentativi fatti.

ESTENSIONI POSSIBILI: Potremmo immaginare una grafica molto migliore con, se possibile, i colori. Potremmo anche fare un gioco con due giocatori, aumentare il numero di finestre, modificare la routine per l'accensione o lo spegnimento...

```
10 REM L'ELETTRICISTA PAZZO
20 M=0
30 DIM A(10):FOR I=1 TO 10:A(I)=0:NEXT I
40 GRAPHICS 0
50 SETCOLOR 2,8,2:SETCOLOR 1,3,6
60 FOR I=1 TO 7
70 POSITION 2,I:FOR J=1 TO 36:PRINT CHR$(160);:NEXT J
80 NEXT I
90 T=0
100 FOR I=1 TO 10
110 T=T+A(I)
120 POSITION 2*I+8,3:PRINT "*"
130 IF A(I)=0 THEN POSITION 2*I+8,3:PRINT " "
140 POSITION 2*I+8,6:PRINT I-1
150 NEXT I
160 IF T=10 THEN 300
170 POSITION 0,14:PRINT "Interruttore numero:";:INPUT N
180 N=N+1
190 M=M+1
200 IF A(N)=1 THEN 280
210 A(N)=1
220 R=RND(1)+N/RND(1)-N:R=(SIN(R)/COS(R))-SIN(RND(1)/N)+336*SIN(8*N)
230 N=INT(10*(R-INT(R)))
240 IF N=0 THEN N=INT(RND(1)*10+1)
250 IF A(N)=1 THEN 280
260 A(N)=1
270 GOTO 90
280 A(N)=0
290 GOTO 220
300 PRINT :PRINT "Trovato in ";M;" prove."
```

La corsa al 20

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: logica (due o più giocatori).
Difficoltà del gioco: media.

IL GIOCO: È possibile immaginare per questo gioco varie regole; abbiamo preso qui solo la regola usuale per due giocatori, che ne fa un gioco simile al gioco dei fiammiferi o al gioco di spostamento: 4 mucchi di 20 stelle appariranno all'inizio della partita ed ogni giocatore può prendere tante stelle quante ne desidera, ma in un solo mucchio per volta. Quello che prende l'ultima stella è considerato come perdente.

IL PROGRAMMA: Il ciclo I (30-50) disegna 4 file di 20 stelle.
La variabile F, dimensionata alla linea 70, servirà a registrare il numero di stelle prese in ogni fila.
La variabile T conterrà il numero totale di stelle prese (linea 60).
La linea 120 controlla che ci siano sufficienti stelle sulla linea indicata.
Il ciclo I(150-190) e il ciclo J (160-180) levano le stelle; la linea 220 testa la fine della partita.

PER GIOCARE: Lanciare il programma con RUN; si visualizzano il tavolo da gioco e i messaggi GIOCATORE 1 e LINEA, poi CASELLE. Si risponde inserendo un numero di linea (tra 1 e 4) seguito da RETURN, ed il numero di stelle da levare seguito da RETURN. Le stelle sono allora eliminate, e toccherà al secondo giocatore. La partita termina, quando l'ultima lettera è stata tolta, con la visualizzazione del giocatore vincente.

ESTENSIONI POSSIBILI: Oltre ad una miglioria della qualità grafica e dei controlli, numerose varianti sono ipotizzabili modificando se necessario le regole di partenza: possiamo autorizzare la presa solo di un numero dispari di stelle, o alternare pari e dispari, oppure prendere a caso il numero di stelle da levare, o ancora giocare in tanti...

```
10 REM LA CORSA AL 20
20 GRAPHICS 0
30 FOR I=1 TO 4
40 POSITION 0,2*I:PRINT I;"   ********************"
50 NEXT I
60 T=0
70 DIM F(4):FOR I=0 TO 4:F(I)=0:NEXT I
80 B=1
90 POSITION 0,12:PRINT "Giocatore :";B
100 PRINT :PRINT "Linea    :";:INPUT L
110 PRINT :PRINT "Caselle  :";:INPUT C
120 IF C+F(L)>20 THEN 90
130 F(L)=F(L)+C
140 T=T+C
150 FOR I=1 TO 4
160 FOR J=F(I)-C TO F(I)
170 POSITION 24-J,2*I:PRINT " "
180 NEXT J
190 NEXT I
200 B=B+1
210 IF B=3 THEN B=1
220 IF T<80 THEN 90
230 PRINT "Il giocatore ";B;" vince."
240 FOR I=1 TO 11
250 SOUND 0,64,10,(I/2=INT(I/2))*4
260 FOR T=1 TO 40:NEXT T
270 NEXT I
```

Musica

RIFERIMENTI: Programmazione: media.
Categoria del gioco: gioco educativo.
Difficoltà del gioco: media.

IL GIOCO: Si presenta come un esercizio di insegnamento del solfeggio: l'apparecchio annuncia una nota (in chiave di sol), seguita dal messaggio GRAVE o ACUTA, secondo la sua posizione sullo spartito. Si dovrà rispondere indicando su quale linea la nota apparirà.
La nota è allora disegnata sullo spartito, seguita da un messaggio BRAVO o PERSO, secondo che si abbia vinto o meno.

IL PROGRAMMA: Il ciclo I (50-70) disegna lo spartito. Un secondo ciclo I (80-120) visualizza i numeri di linea.
La linea 130 definisce il nome delle note, e la linea 140 realizza la presa casuale di una nota.
Il nome della nota è calcolato alla linea 150 ed è visualizzato dalla linea 160, seguito dal messaggio GRAVE o ACUTA, secondo il caso.
La risposta sarà inserita alla linea 190.
La linea 200 si occupa della decodifica della risposta.
La linea 210 visualizza la nota.
Le linee 230 e 240 guardano se si è vinto o perso e visualizzano il messaggio corrispondente.
La linea 250 permette di fare un nuovo tentativo.

PER GIOCARE: Si lancia il programma con RUN.
Si ottiene allora l'apparizione dello spartito, così come il nome della nota da posizionare e la domanda NUMERO DI LINEA.
Si risponde dando un numero di linea (da 1 a E) seguito da RETURN.
La nota è allora disegnata e il messaggio BRAVO o PERSO appare, in questo caso, sullo schermo.
Si deve battere RETURN per continuare, e fermarsi battendo BREAK.

ESTENSIONI POSSIBILI: Possiamo estendere ad un'altra chiave, che non sia la chiave di sol, si può introdurre il colore, i suoni, ecc.

```
10 REM MUSICA
20 DIM N$(21),L$(1)
30 SETCOLOR 1,5,2:SETCOLOR 2,12,6
40 PRINT "}"
50 FOR I=1 TO 5
60 POSITION 2,2*I+4:FOR J=1 TO 35:PRINT CHR$(18);:NEXT J
70 NEXT I
80 FOR I=1 TO 14
90 POSITION 0,I+2
100 IF I<=9 THEN PRINT I
110 IF I>9 THEN PRINT CHR$(I+55)
120 NEXT I
130 N$="SI LA SOL FA MI RE DO "
140 X=INT(RND(1)*14+1)
150 T=X-1-7*INT((X-1)/7)
160 POSITION 0,18:PRINT "Nota ";N$(3*T+1,3*T+3);
170 IF X<=7 THEN PRINT " acuta."
180 IF X>7 THEN PRINT " grave."
190 POSITION 0,20:PRINT "Numero di linea :";:INPUT L$
200 L=(ASC(L$)-48)*(L$<":")+(ASC(L$)-55)*(L$>="A")
210 POSITION 14,X:PRINT CHR$(2):POSITION 14,X+1:PRINT CHR$(2):POSITION 14,X+2:PR
INT CHR$(96)
220 POSITION 25,20
230 IF L=X THEN PRINT "BRAVO."
240 IF L<>X THEN PRINT "PERSO!"
250 PRINT :PRINT "Battere <RETURN> per continuare ";:INPUT L$
260 GOTO 40
```

Cavallina

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: bassa.

IL GIOCO: Anche se l'idea è in sé molto semplice, non sarà obbligatoriamente evidente scoprire una strategia adeguata. Disponiamo di tre più tre pecore, raffigurate con dei simboli grafici, che occupano 8 caselle; le pecore possono spostarsi da una casella ad un'altra libera. Varie regole potranno intervenire, secondo la vostra stima personale su questa o quella: o si costringono le pecore ad andare sulla prima casella disponibile, saltando eventualmente sopra le pecore avversarie, oppure si lascia loro scegliere la casella di arrivo, il che può rendere le partite interminabili; a voi la scelta.

IL PROGRAMMA: Il ciclo I (50-80) inizializza un array A(I), che contiene le posizioni delle pecore. Il ciclo I (100-150) disegna il campo di gioco e le pecore.
Le linee 170 e 180 si occupano del testo e degli ingressi.
La linea 190 controlla la validità del colpo. Il cambio giocatore è realizzato alla linea 220.

PER GIOCARE: Si comincia l'esecuzione del programma con RUN. Si ottiene allora il messaggio GIOCATORE 1, che indica il numero del giocatore, seguito dalla domanda INIZIO, FINE?
Si risponde indicando il numero della pecora da spostare seguito da RETURN, poi il numero della casella di arrivo, seguito da RETURN.
È allora realizzato lo spostamento; poi il secondo giocatore procede nello stesso modo. Il vincitore è in teoria quello che riesce a posizionare le sue tre pecore allineate sulle posizioni inizialmente occupate dall'avversario.

ESTENSIONI POSSIBILI: Si comincerà con l'aggiungere un controllo del giocatore vincente.
Si può anche migliorare la presentazione disegnando delle vere pecore, rendere il gioco più attraente numerandole e imponendo un certo ordine all'arrivo. Infine potremmo creare una strategia per il calcolatore, fissando le regole.

```
10 REM CAVALLINA
20 GRAPHICS 0
30 DIM A(8)
40 A(1)=0:A(8)=0
50 FOR I=1 TO 3
60 A(I+1)=1
70 A(I+4)=-1
80 NEXT I
90 H=1
100 FOR I=1 TO 8
110 POSITION 4*I,8:PRINT I
120 POSITION 4*I-1,6:PRINT "   "
130 IF A(I)=1 THEN POSITION 4*I-1,6:PRINT CHR$(26);CHR$(21);CHR$(10)
140 IF A(I)=-1 THEN POSITION 4*I-1,6:PRINT CHR$(8);CHR$(21);CHR$(3)
150 NEXT I
160 J=1+(H=-1)
170 POSITION 0,13:PRINT "Giocatore ";J
180 PRINT :PRINT "Inizio , Fine :               ";:POSITION 18,15:INPUT D,F
190 IF A(D)<>H OR A(F)<>0 THEN 170
200 A(D)=0
210 A(F)=H
220 H=-H
230 GOTO 100
```

Lettere

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO: È ispirato da un gioco televisivo molto noto in Francia. L'ATARI sceglie a caso un certo numero di simboli (vocali e consonanti) nell'alfabeto e ognuno dei giocatori dovrà indicare la lunghezza della parola che è riuscito ad ottenere (l'apparecchio non verificherà se la parola è corretta!).
Il punteggio totale di ogni giocatore è visualizzato in seguito.

IL PROGRAMMA: Il ciclo I (100-130) sceglie nell'alfabeto tra 7 e 12 lettere, in funzione della variabile casuale L, presa alla linea 70. Per evitare di avere troppe consonanti, si utilizza in linea 110 una piccola astuzia, che consiste nel rifare a caso una presa nell'array A$, contenente delle vocali (se ne può avere più o meno utilizzando delle altre RND).
Il ciclo 140-190 chiede ad ognuno dei due giocatori la lunghezza della loro parola.
Le linee 200 e 210 paragonano i risultati e totalizzano il punteggio di ogni giocatore, visualizzato alla linea 220.

PER GIOCARE: Si lancia l'esecuzione del gioco con RUN.
Appaiono le lettere, poi la domanda GIOCATORE 1 LUNGHEZZA? alla quale il primo giocatore risponde dando la lunghezza della parola che ha ottenuto (la validità della parola dovrà essere tuttavia controllata dall'altro giocatore); poi il giocatore 2 farà altrettanto ed il punteggio di ogni giocatore sarà visualizzato.
Per continuare la partita, basterà battere RETURN.
Si dovrà battere BREAK per finire.

ESTENSIONI POSSIBILI: Si potrebbe naturalmente pensare di introdurre una temporizzazione prima di chiedere le risposte ai giocatori.
Si potrebbe anche introdurre non la lunghezza della parola, ma la parola stessa, l'avversario dovrà allora rispondere alla domanda CORRETTA O NO? Si potrebbe anche estendere il gioco a più di due giocatori. E perché non cercare di definire un gioco contro l'ATARI, nel quale sarà stata inserita prima una lista abbastanza lunga di parole, di cui visualizzerebbe le lettere disordinatamente, con eventualmente qualche lettera di complemento scelta a caso.

```
10 REM LETTERE
20 DIM A$(7),R$(1)
30 S=0:J=0
40 GRAPHICS 2
50 SETCOLOR 0,10,2
60 PRINT "}"
70 L=INT(RND(1)*6+7)
80 A$="AEIOUEA"
90 POSITION 0,3:PRINT #6;"LETTERE:"
100 FOR I=1 TO L
110 POSITION 7+I,3:PRINT #6;CHR$(INT(RND(1)*26+65))
120 IF RND(1)<0.4 THEN POSITION 7+I,3:R=INT(RND(1)*7+1):PRINT #6;A$(R,R)
130 NEXT I
140 I=1
150 PRINT "  Giocatore ";I;:PRINT " - Lunghezza :";:INPUT L
170 IF I=1 THEN P=L
180 I=I+1
190 IF I<3 THEN 150
200 S=S+P*(P>L)
210 J=J+L*(L>P)
220 PRINT "   Punti - Giocatore 1 :";S;" -Giocatore 2 :";J
230 PRINT "  Battere <RETURN> per continuare";:INPUT R$
240 GOTO 40
```

Macchina delle parole

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco d'azzardo (sulle parole).
Difficoltà del gioco: nessuna.

IL GIOCO: Questo cortissimo programma permette, all'ATARI, di dimostrare le sue capacità di manipolazione di stringhe di caratteri: si parte semplicemente da un prefisso e da un suffisso di origine greca, che assembleremo a caso, cadremo naturalmente ogni tanto su delle parole assurde o anche leggermente equivoche. Una nuova parola è confezionata ad ogni tentativo, ma è possibile ottenere varie volte la stessa parola.

IL PROGRAMMA: Due variabili A$ e R$ sono dimensionate alle linee 30 e 40: una conterrà i prefissi, poi i suffissi; l'altra la parola composta. Il ciclo J legge la linea DATA 1000 riguardante i prefissi, poi la linea DATA 2000 per i suffissi. La linea 80 effettua la scelta di una fila a caso. Nei DATA la parola ottenuta alla linea 110 è visualizzata dalla linea 130, poi si ricomincia (linea 140).

PER GIOCARE: Lanciare l'esecuzione del programma con RUN. Una prima parola sarà visualizzata immediatamente, poi una nuova parola apparirà al seguito della precedente.
Battere BREAK per uscire dal programma.

ESTENSIONI POSSIBILI: Si potrà facilmente aumentare la lista delle parole disponibili.
Potremmo anche utilizzare un processo analogo per fare creare delle frasi al calcolatore, fargli inventare delle poesie, ecc. Potremmo anche trasformare il programma iniziale in un gioco di società che consisterebbe, per esempio, nel ricevere delle scommesse sulla prossima parola che apparirà. E affidare, eventualmente, la gestione delle scommesse all'ATARI...

```
10 REM MACCHINA DELLE PAROLE
20 GRAPHICS 0
30 DIM A$(6)
40 DIM R$(12)
50 R$=""
60 FOR J=1 TO 2
70 RESTORE 1000*J
80 FOR R=1 TO RND(1)*15+1
90 READ A$
100 NEXT R
110 R$(LEN(R$)+1)=A$
120 NEXT J
130 PRINT R$
140 GOTO 50
1000 DATA ANDRO,COSMO,MICRO,ELIO,SESSO,STEREO,PALEO,MACRO,IDRO,STENO,PATO,OLEO,M
ITO,OMEO,OTTO
2000 DATA CRATE,FILO,FOBO,TROPO,GIMNO,MANE,NAUTA,FAGO,FUGO,PATO,CARDIO,DERMA,SCO
PIO,VORO,FONO
```

Spia

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: bassa.

IL GIOCO: Il programma può servire simultaneamente da codifica e decodifica di messaggi segreti, seguendo un metodo molto facile da mettere in opera su questo tipo di calcolatore.
Un piccolo generatore casuale situato alla fine del programma è utilizzato per modificare in maniera arbitraria ogni lettera del messaggio; si dovrà fare attenzione ad utilizzare sempre lo stesso numero (nucleo) per cominciare la codifica o la decodifica di due messaggi identici.

IL PROGRAMMA: Il nucleo iniziale (che sarà un intero qualunque) è inserito alla linea 40; il messaggio è inserito alla linea 60.
La linea 70 permette di scegliere tra le opzioni di codifica e decodifica.
Il ciclo I (90-160) è utilizzato sia per effettuare la codifica, che la decodifica.
La linea 110 chiama il generatore casuale.
La linea 120 realizza la trasposizione delle lettere.
Le linee 130 e 140 verificano che si resti nei limiti autorizzati.
I risultati saranno stampati dalla linea 170.
Le linee da 1000 a 1030 contengono il generatore.

PER GIOCARE: Lanciare il programma con RUN.
Inserire in seguito un intero N che servirà da nucleo per il generatore (bisognerà ricordarselo per la decodifica).
Poi battere il messaggio, seguito da RETURN.
Scegliere l'opzione C per CODIFICARE, e l'opzione D per DECODIFICARE, e battere RETURN.
Il messaggio codificato o decodificato secondo il caso, sarà allora visualizzato.

ESTENSIONI POSSIBILI: Si potrà estendere la codifica a segni diversi dalle lettere dell'alfabeto, utilizzare eventualmente vari generatori, introdurre nel messaggio una codifica del o dei numeri che sono serviti da generatori, ecc.

```
10 REM SPIA
20 DIM A$(300),B$(300),R$(1)
30 GRAPHICS 0
40 PRINT "Fornisci il codice :";:INPUT N
50 A=1/N
60 PRINT :PRINT "Fornisci il messaggio :";:INPUT A$
70 PRINT :PRINT "Vuoi (C)odificare o (D)ecodificare ";:INPUT R$
80 B$=""
90 FOR I=1 TO LEN(A$)
100 T=ASC(A$(I,I))
110 GOSUB 1000
120 T=T-A*((R$="D")-(R$="C"))
130 IF T>90 THEN T=T-26
140 IF T<65 THEN T=T+26
150 B$(LEN(B$)+1)=CHR$(T)
160 NEXT I
170 PRINT :PRINT "Risultato :";B$:END
1000 A=(A+3.14159)^8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN
```

Attaccate i vagoni

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: solitario.
Difficoltà del gioco: bassa.

IL GIOCO: Bisogna cercare di agganciare, il più rapidamente possibile, la locomotiva e il suo vagone, che appaiono tutti e due all'inizio della partita, in due posti più o meno lontani dello schermo. Dovremo per questo spostare la locomotiva indicando la sequenza dei movimenti da effettuare (in maniera codificata: A per una casella verso l'alto; B per una casella verso il basso; D per una casella verso destra; e S per una casella verso sinistra).
Si batterà, per esempio: BBSDAAA.

IL PROGRAMMA: Le linee 50 e 70 disegnano la locomotiva, la linea 80 il vagone, la linea 90 le ruote.
La posizione iniziale della locomotiva è definita dalle linee 110 e 120, quella del vagone dalle linee 130 e 140.
La linea 150 evita che siano inizialmente troppo vicini.
Le linee 210 e 220 realizzano la visualizzazione del vagone, le linee 170-200 la locomotiva.
Il test di fine è in linea 230.
I movimenti da effettuare sono inseriti in una variabile alfanumerica R$, alla linea 240, e sono decodificati in un ciclo I (270-300), per essere eseguiti alla linea 310.

PER GIOCARE: RUN permette di visualizzare le posizioni di partenza della locomotiva e del vagone.
Si batte allora la serie degli spostamenti da effettuare, sotto la forma DDDAAABASS, seguito da RETURN.
La locomotiva apparirà nella sua nuova posizione, e si ricomincerà, fino a che non sia attaccata al vagone.
Si otterrà allora l'apparizione del punteggio.

ESTENSIONI POSSIBILI: Sarà interessante prevedere la perdita della partita se per incidente, nel corso dello spostamento, la locomotiva entra in collisione col vagone.
Potremmo anche prevedere vari vagoni, e un ordine di aggancio, giocare in molti con varie locomotive, disegnare le ferrovie, introdurre degli ostacoli da aggirare obbligatoriamente da una certa parte, ecc.

```
10 REM ATTACCATE I VAGONI
20 GRAPHICS 0
30 SETCOLOR 1,3,2:SETCOLOR 2,12,8
40 DIM A$(5),B$(5),C$(5),D$(5),E$(5),R$(40)
50 A$="re w ":REM battere le lettere minuscole schiacciando simultaneamente<CONT
ROL>
60 B$=" arxe":REM vedere 40
70 C$="rxrrc":REM vedere 40
80 D$="zrrrc":REM vedere 40
90 E$=" O O"
100 N=0
110 LY=INT(RND(1)*36)
120 LX=INT(RND(1)*19)
130 WY=INT(RND(1)*31)
140 WX=2+INT(RND(1)*19)
150 IF ABS(LX-WX)<4 THEN 110
160 PRINT "}"
170 POSITION LY,LX:PRINT A$
180 POSITION LY,LX+1:PRINT B$
190 POSITION LY,LX+2:PRINT C$
200 POSITION LY,LX+3:PRINT E$
210 POSITION WY,WX:PRINT D$
220 POSITION WY,WX+1:PRINT E$
230 IF LX=WX-2 AND WY=LY-5 THEN 400
240 POSITION 0,22:PRINT "Percorso da seguire (DSAB....)";:INPUT R$
250 X=0:Y=0
260 N=N+1
270 FOR I=1 TO LEN(R$)
280 X=X+(R$(I,I)="B" AND LX<18)-(R$(I,I)="A" AND LX>0)
290 Y=Y+(R$(I,I)="D" AND LY<35)-(R$(I,I)="S" AND LY>0)
300 NEXT I
310 LX=LX+X:LY=LY+Y
320 GOTO 160
400 POSITION 6,22:PRINT "BRAVO  ! riuscito in ";N;" colpi.";
410 INPUT R$
```

Lancio di un dado truccato

RIFERIMENTI: Programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: media.

IL GIOCO: Riprende un'idea di un programma precedente (DADO TRUCCATO) in una applicazione statistica: si fabbrica un dado volontariamente truccato, che si lancerà un certo numero di volte, e si crea una tabella di risultati.

IL PROGRAMMA: Le probabilità desiderate per ogni tiro sono introdotte da un ciclo I (50-120), con tutte le verifiche di validità necessarie. Il numero di tiri richiesti è introdotto alla linea 140.
Il ciclo J (150-230) contabilizza i lanci.
Il ciclo I (170-190) assicura la conformità di ogni lancio con le probabilità introdotte all'inizio del gioco.
I risultati sono contabilizzati alla linea 200, e visualizzati grazie alle linee 210 e 220.

PER GIOCARE: Cominciare l'esecuzione battendo RUN, poi, alla domanda PROBABILITÀ DEL DADO 1, introdurre un numero tra 0 e 1.
Fare lo stesso per i dadi 2, 3, ecc., senza dimenticare che il totale delle probabilità introdotte dovrà essere vicinissimo a 1.
In seguito, all'apparizione della domanda NUMERO DI LANCI si introdurrà un intero sufficientemente grande, seguito da RETURN. La visualizzazione dei risultati richiederà tanto più tempo ad apparire quanto più è grande l'intero inserito.
I risultati sono visualizzati sotto i numeri dei lanci: così 32.3 apparso sotto l'1 significa che il lancio di 1, al quale avrete assegnato una probabilità di 0.3 è stata realizzata 32 volte.

ESTENSIONI POSSIBILI: Possiamo immaginare di migliorare la raffigurazione dei risultati utilizzando per esempio un istogramma, oppure un qualsiasi altro metodo graficamente più soddisfacente.

```
10 REM LANCIO DI UN DADO TRUCCATO
20 DIM E(6)
30 GRAPHICS 0
40 P=0
50 FOR I=1 TO 6
60 POSITION 0,3:PRINT "                                    "
70 POSITION 0,3:PRINT "Probabilita' del dado ";I;:INPUT E:E(I)=E
80 P=P+E(I)
90 IF E(I)<0 OR E(I)>=1 OR P>1.1 THEN 300
100 POSITION 6*I-4,6:PRINT "DADO ";I
110 POSITION 6*I-3,8:PRINT E(I)
120 NEXT I
130 IF P<0.9 THEN RUN
140 POSITION 0,11:PRINT "Numero di lanci :";:INPUT P
150 FOR J=1 TO P
160 N=INT(RND(1)*6+1)
170 FOR I=1 TO 6
180 IF N=I AND RND(1)>E(I)-INT(E(I)) THEN 160
190 NEXT I
200 E(N)=E(N)+1
210 POSITION 6*N-3,15:PRINT N
220 POSITION 6*N-4,17:PRINT E(N)
230 NEXT J
240 END
300 POSITION 0,4:PRINT "IMPOSSIBILE!!! :0<=P<=1"
310 P=P-E(I)
320 FOR J=1 TO 1700:NEXT J
330 POSITION 0,4:PRINT "                         "
340 GOTO 60
```

Gioco con le carte

RIFERIMENTI:

Programmazione: difficile.
Categoria del gioco: gioco di prestidigitazione.
Difficoltà del gioco: bassa.

IL GIOCO:

È un gioco abbastanza facile ma, qui, le carte sono manipolate sotto i vostri occhi dall'ATARI che ha il ruolo di prestigiatore; 21 carte, prese a caso in un mazzo di 52, sono visualizzate, e il giocatore dovrà scegliere mentalmente una delle carte proposte. Poi dovrà indicare all'ATARI in quale colonna (1ma, 2nda, o 3za partendo da sinistra) si trova la carta scelta; le carte sono allora ridistribuite ed il giocatore indica di nuovo la colonna corrispondente alla sua carta; questo è fatto di nuovo ancora una volta, alla quarta visualizzazione, l'ATARI è in grado di indovinare la carta scelta.

IL PROGRAMMA:

Il ciclo I (60-120) permette di scegliere, a caso, 21 carte su 52, ed il ciclo chiuso J (80-100) permette di assicurarsi che non si realizzi due volte la stessa scelta.
La linea 140 contiene il colore delle carte, così come il nome delle figure (FA, DO, RE, AS), in una variabile alfanumerica B$.
Il ciclo I (150-240) realizza la visualizzazione del gioco.
La linea 160 decodifica il colore della carta.
La linea 170 sceglie la posizione di visualizzazione.
La linea 180 decodifica l'altezza.
La visualizzazione vera e propria è realizzata dalla linea 190 alla linea 230.
La linea 260 permette, al giocatore, di inserire il numero della colonna dove si trova la carta. Il ciclo J (300-320) realizza la mescola, ed i cicli chiusi J (350-390) e I (360-380) fabbricano la nuova variabile A$ da utilizzare per la prossima visualizzazione.

PER GIOCARE:

Si lancia il programma con RUN.
Appaiono le prime 21 carte, divise in tre colonne.
Si indica allora il numero (1, 2, 3) della colonna dove si trova la carta scelta, seguito da RETURN.
Si ottiene una seconda visualizzazione e si procede nella stessa maniera; poi una terza e si fa la stessa cosa.
Alla quarta mano, la carta scelta è automaticamente visualizzata.
Si uscirà dal programma battendo BREAK al posto del numero di colonna.

ESTENSIONI POSSIBILI: Si può immaginare una miglioria della visualizzazione scrivendo il nome delle carte per intero, o anche disegnando delle vere carte, a colori.
Si potrà anche cercare di programmare degli altri giochi, usando lo stesso principio di base, ma con più carte, ecc.

```
10 REM GIOCO CON LE CARTE
20 GRAPHICS 0
30 SETCOLOR 1,7,2:SETCOLOR 2,15,8:SETCOLOR 4,15,8
40 DIM A$(21),B$(21),R$(5)
50 A$=""
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND(1)*52+38))
75 IF A$="" THEN 110
80 FOR J=1 TO LEN(A$)
90 IF B$=A$(J,J) THEN 70
100 NEXT J
110 A$(LEN(A$)+1)=B$
120 NEXT I
130 H=3:P=1
140 B$="FIQUCUPIFADOREAS"
150 FOR I=1 TO LEN(A$)*P+11*(P=0)
160 S=INT((ASC(A$(I,I))-38)/13)
170 J=7*INT((I-1)/7)
180 T=ASC(A$(I,I))-37-13*S
190 IF P=0 THEN POSITION 0,0:PRINT "La tua carta e' :                    "
200 POSITION 4+J*P+INT(J/2)+16*(P=0),(I-J)*P
210 IF T<=9 THEN PRINT STR$(T+1);
220 IF T>9 THEN B=ABS(2*T-11):PRINT B$(B,B+1);
230 POSITION 4+(J+3)*P+1+INT(J/2)+21*(P=0),(I-J)*P:PRINT B$(2*S+1,2*S+2)
240 NEXT I
250 IF P=0 THEN END
260 POSITION 0,16:PRINT "Quale colonna :";:INPUT N
270 PRINT "}"
280 H=H-1
290 R$="20120":B$=""
300 FOR J=0 TO 2
310 B=7*VAL(R$(N+J,N+J))+1:B$(LEN(B$)+1)=A$(B,B+6)
320 NEXT J
330 IF H=0 THEN A$=B$:P=0:GOTO 140
340 A$=""
350 FOR J=0 TO 2
360 FOR I=1 TO 7
370 A$(LEN(A$)+1)=B$(3*I-2+J,3*I-2+J)
380 NEXT I
390 NEXT J
400 GOTO 140
```

Il gioco di Marienbad

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: logica (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO: Un gioco reso celebre grazie ad un film viene presentato qui in una versione più conforme alle sue origini, ossia il gioco di NIM.
Infatti, è possibile, all'inizio della partita, scegliere il numero di fiammiferi che vogliamo mettere in ogni mucchio.
Ogni giocatore gioca turno che prende l'ultimo fiammifero ha perso.

IL PROGRAMMA: La linea 70 dimensiona una variabile A(I) che conterrà il numero di fiammiferi di ogni mucchio.
Il ciclo I (80-110) permette di inserire questo numero.
Il ciclo I (140-170) presenta il mucchio.
La linea 180 garantisce la visualizzazione del numero del giocatore.
La linea 200, così come la 210, controllano se la mossa è autorizzata.
La linea 280 controlla la fine della partita.

PER GIOCARE: Si lancia il programma con RUN, poi si inserisce il numero di mucchi (tra 1 e 7), seguito da RETURN.
Si inserirà in seguito il numero di fiammiferi in ogni mucchio (fare ad ogni ingresso N seguito da RETURN). Si otterrà allora la visualizzazione dei differenti mucchi, col numero del giocatore interessato (1 o 2), così come il messaggio IN QUALE MUCCHIO?; si risponde battendo il numero del mucchio nel quale si desidera pescare seguito da RETURN. Si ottiene allora la domanda QUANTITÀ DA TOGLIERE?, e si risponde indicando il numero di fiammiferi che si desidera levare dal mucchio.
Poi sarà il turno del secondo giocatore, e si giocherà alternativamente fino a che non ci siano più fiammiferi disponibili.
In fine partita, si otterrà la visualizzazione del numero del giocatore vincente.

ESTENSIONI POSSIBILI: Si può designare ogni giocatore col suo nome, fare una grafica più elaborata, scrivere un programma che permetta all'ATARI di fare da partner, ecc.

```
10 REM IL GIOCO DI MARIENBAD
20 GRAPHICS 0
30 DIM N$(1)
40 PRINT "Numero di mucchi (<=7):";:INPUT N
50 IF N<=0 OR N>=8 THEN RUN
60 S=0
70 DIM A(N)
80 FOR I=1 TO N
90 PRINT :PRINT "Mucchio ";I;" - numero di fiammiferi :";:INPUT A:A(I)=A
100 S=S+A(I)
110 NEXT I
120 H=1
130 PRINT "}"
140 FOR I=1 TO N
150 POSITION 5*I+2,4:PRINT A(I)
160 POSITION 0,6:PRINT "Mucchio":POSITION 5*I+2,6:PRINT I
170 NEXT I
180 POSITION 2,10:PRINT "giocatore ";H
190 PRINT :PRINT "In che mucchio          :";:INPUT T
200 IF T>N THEN 130
210 IF A(T)=0 THEN 130
220 POSITION 2,14:PRINT "Quantita' da togliere :      ";:POSITION 25,14:INPUT W
230 IF W>A(T) OR W=0 THEN 210
240 A(T)=A(T)-W
250 S=S-W
260 H=H+1
270 IF H=3 THEN H=1
280 IF S=0 THEN PRINT :PRINT "Il giocatore ";H;" vince.":END
290 GOTO 130
```

L'indesiderato

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: fortuna e riflessione (1 giocatore).
Difficoltà del gioco: media.

IL GIOCO: Siete proprietario di un immobile con 16 appartamenti vuoti e non riuscite a mettere le mani sul vostro inquilino indesiderato. Visitate gli appartamenti a caso ma, se visitate l'appartamento vicino, il vostro inquilino sentendo il rumore, si sposta. Può anche installarsi in un appartamento che avete già visitato e segnato come vuoto. Per vincere bisognerà mandarlo via con meno di 9 mosse.

IL PROGRAMMA: Un array B è dimensionato a 16, alla linea 20, per ricevere i numeri delle caselle già visitate.
Il ciclo I (80-110) disegna il campo di gioco.
La posizione iniziale dell'inquilino è scelta a caso dalla linea 120.
Il numero della casella da visitare è dato dalla linea 130.
Le linee da 135 a 220 controllano le risposte possibili.
Il ciclo I (180-200) disegna le caselle visitate.
La linea 230 disegna l'inquilino, cancellato alla linea 260.
Le linee 270 e 280 assicurano il suo trasferimento.
Le linee da 500 a 1000 assicurano la decodifica della musica da suonare.

PER GIOCARE: Si lancia il programma con RUN.
Alla domanda CASELLA DA VISITARE? si risponde battendo una lettera da A a P. Se si è trovato l'inquilino, questi viene visualizzato e la partita è finita; se non siete riusciti a trovarlo si possono verificare due casi: se siamo abbastanza lontani e l'inquilino non si sposta, oppure è in una casella e appare per un breve istante, e si nasconde a caso in una casella vicina. Nelle caselle già visitate appare un cuore e non potrete più ritornarci. Se, dopo 9 colpi, non l'avete trovato, la posizione dell'inquilino si evidenzia e avete perso.

ESTENSIONI POSSIBILI: Si potrà certamente giocare con più di 16 caselle, utilizzare dei caratteri grafici per rappresentare l'inquilino e il proprietario, utilizzare un algoritmo più fine per lo spostamento dell'inquilino, vietargli eventualmente di ritornare nelle caselle già visitate, ecc.

```
10 REM L'INDESIDERATO
20 DIM A$(1),B$(3),C$(1),M$(8),NO$(2),B(16)
25 FOR I=1 TO 16:B(I)=0:NEXT I
30 A$=CHR$(160)
40 FOR I=1 TO 3:B$(LEN(B$)+1)=A$:NEXT I
50 GRAPHICS 0
60 SETCOLOR 1,3,2:SETCOLOR 2,8,10
70 PRINT ")"
80 FOR I=1 TO 16
90 POSITION 4+4*(I-1-4*INT((I-1)/4)),2+4*INT((I-1)/4):PRINT B$
100 POSITION 4+4*(I-1-4*INT((I-1)/4)),3+4*INT((I-1)/4):PRINT A$;:PRINT CHR$(64+I
);:PRINT A$
110 NEXT I
120 X=INT(RND(1)*16+1):S=1:K=0
130 C$="":POSITION 0,20:PRINT "Casella da visitare :      ";:POSITION 21,20:INPUT
 C$:C=ASC(C$)-64
135 IF C<1 OR C>16 THEN 130
140 IF B(C)=1 THEN POSITION 0,22:PRINT "Gia' visitata!":GOTO 130
150 IF C=X THEN POSITION 0,22:PRINT "Scovato in ";S;" colpi.":M$="DOMISO":GOSUB
500:K=1:GOTO 230
160 IF ABS(X-C)<=5 AND ABS(X-C)<>2 THEN 230
170 POSITION 0,22:PRINT "Non e' la'!":M$="DOREMI":GOSUB 500
180 S=S+1:B(C)=1:FOR I=1 TO 16
190 POSITION 4+4*(I-1-4*INT((I-1)/4)),2+4*INT((I-1)/4):PRINT CHR$(ASC(B$)-32*B(I
))
200 NEXT I
210 IF S=9 THEN POSITION 0,22:PRINT "Hai perso, era in ";CHR$(X+64);".":K=1:GOTO
 230
220 POSITION 0,22:PRINT "                                           ":GOTO 130
230 POSITION 5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X-1)/4):PRINT "%"
240 FOR I=1 TO 300:NEXT I
250 IF K=1 THEN M$="SOMIDO":GOSUB 500:POSITION 0,23:END
260 POSITION 5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X-1)/4):PRINT " "
270 Z=INT(10*RND(1)+1)-5:IF ABS(Z)=2 THEN 270
280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO 270
290 GOTO 180
500 FOR I=1 TO LEN(M$)/2
510 READ NO$,NO:IF NO$<>M$(2*I-1,2*I) THEN 510
520 SOUND 0,NO,10,4
530 FOR T=0 TO 30:NEXT T
540 RESTORE
550 NEXT I
560 RETURN
1000 DATA DO,126,RE,108,MI,96,FA,91,SO,81,LA,72,SI,64
```

Componi

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: composizione musicale.
Difficoltà del gioco: nessuna.

IL GIOCO: Si tratta, infatti, di un programma capace di comporre, in modo molto elementare, della musica, e questo in una maniera completamente casuale. Il calcolatore, in un primo tempo, disegnerà uno spartito, poi stamperà una successione di note (che vanno dal DO grave al SI acuto) e assegnerà loro il nome. Una nuova serie di note apparirà in seguito, e così via.

IL PROGRAMMA: Il ciclo I (50-70) disegna lo spartito.
La variabile N$, alla linea 80, contiene i nomi delle note e VA$ alla linea 35 contiene le note in forma codificata.
Il secondo ciclo I (90-170) permette di fare la scelta di una nota in maniera casuale, alla linea 100; il suo nome è calcolato alla linea 110 e la nota è disegnata alla linea 120.
La linea 180 permette di ricominciare battendo RETURN.

PER GIOCARE: Lanciare l'esecuzione del programma battendo RUN.
Lo spartito sarà visualizzato rapidamente, poi le note saranno rappresentate col loro nome in basso sullo schermo.
Una pressione su RETURN provocherà l'apparizione di una nuova serie di note, ed una pressione sul tasto BREAK provoca la fine della partita.

ESTENSIONI POSSIBILI: Potremmo affinare le regole di composizione, visualizzare più note, introdurre dei diesis e dei bemolle, migliorare la grafica , fare suonare la melodia visualizzata.

```
10 REM COMPONI
20 DIM N$(21),D$(3),VA$(42)
30 SETCOLOR 1,7,2:SETCOLOR 2,13,10:SETCOLOR 4,13,10
35 VA$="031035040045047053063064072081091096108126"
40 PRINT "}"
50 FOR I=1 TO 5
60 POSITION 1,3+2*I:FOR L=1 TO 37:PRINT CHR$(18);:NEXT L
70 NEXT I
80 N$="SI LA SOLFA MI RE DO "
90 FOR I=1 TO 13
100 X=INT(RND(1)*14+1)
110 D=3*(X-7*(X>7))-2:D$=N$(D,D+2)
120 POSITION 3*I-1,X-1:PRINT CHR$(22):POSITION 3*I-1,X:PRINT CHR$(22):POSITION 3
*I-2,X+1:PRINT CHR$(96)
140 POSITION 3*I-2,18:PRINT D$(1,2)
150 D=D+3*7*(X>7)
160 SOUND 0,VAL(VA$(D,D+2)),10,4
165 FOR T=1 TO 50:NEXT T
170 NEXT I
180 POSITION 3,21:PRINT "Battere <RETURN> per continuare";:INPUT D$
190 GOTO 40
```

Giorno della settimana

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: utilità.
Difficoltà del gioco: bassa.

IL GIOCO: Il programma vi permette di ritrovare, senza alcun calcolo, il giorno della settimana corrispondente ad una certa data.
La data sarà inserita sotto forma di tre numeri, per esempio 11, 3, 1956 per l'11 marzo 1956, e apparirà il risultato.

IL PROGRAMMA: Le linee da 30 a 70 controllano gli inserimenti. La linea 40 si occupa della fine del programma. La linea 90 permette una correzione per le date precedenti il 1582 (messa in opera dell'attuale calendario).
Le linee da 100 a 110 permetteranno la visualizzazione in chiaro del mese nella risposta (su 4 lettere).
La formula utilizzata appare alle linee 130 e 140.
Le linee 150 e 160 permetteranno la visualizzazione del giorno nella risposta.
La linea 170 stampa i risultati.
Il sottoprogramma 200-220 interviene per considerare i mesi di gennaio e febbraio.

PER GIOCARE: Lanciare il programma con RUN, poi inserire il giorno seguito da RETURN, il mese (tra 1 e 12) seguito da RETURN, e l'anno (nella forma 1845, per esempio) seguito da RETURN.
Il risultato è rapidamante visualizzato.
Battendo un tasto qualunque ritorna sulla domanda GIORNO?
Per continuare basta battere un giorno, un mese, un anno.
Per terminare, basta rispondere 0 alla domanda GIORNO?

ESTENSIONI POSSIBILI: Si potrà inserire il mese in lettere e fare visualizzare il nome del giorno per intero. Si può immaginare di fare apparire le date delle feste particolari (Natale, Pasqua, Pentecoste...).
Si può scrivere un programma che visualizzerebbe i giorni festivi per un dato anno, ecc.

```
10 REM GIORNO DELLA SETTIMANA
20 GRAPHICS 0
25 DIM M$(48),J$(63)
30 PRINT "Giorno..........:";:INPUT J
40 IF J=0 THEN END
50 PRINT :PRINT "Mese............:";:INPUT M
60 PRINT :PRINT "Anno............:";:INPUT A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="GEN FEB MAR APR MAG GIU LUG AGO SET OTT NOV DIC "
110 M$=M$(4*M-3,4*M)
120 IF M<3 THEN GOSUB 200
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DOMENICA LUNEDI   MARTEDI  MERCOLEDIGIOVEDI  VENERDI  SABATO   "
160 J$=J$(9*S+1,9*S+9)
170 POSITION 2,15:PRINT "Il ";J;" ";M$;" ";B;" era un ";J$;"."
180 POSITION 3,23:PRINT "Battere <RETURN> per continuare";:INPUT M$
190 RUN
200 M=M+12
210 A=A-1
220 RETURN
```

Gioco della vita

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco solitario (2 versioni).
Difficoltà del gioco: bassa.

IL GIOCO: La versione presentata qui rispetta le regole poste all'origine dal matematico J.H. CONWAY.
Una certa popolazione è definita all'inizio del gioco sia dall'ATARI (versione automatica) sia da voi stessi (versione manuale). Questa popolazione evolverà sotto i vostri occhi, rispettando certe regole, molto elementari di nascita, di vita e di morte. L'evoluzione proseguirà fino a che non si deciderà di fermare il programma.

IL PROGRAMMA: Due array (B e C) sono utilizzati alla linea 30, per registrare lo stato della popolazione. I cicli F (230-300) e G (240-290) sono utilizzati all'inizio del gioco per definire la popolazione iniziale (linee 250 e 260). Il calcolo della nuova popolazione è realizzato in seguito con l'aiuto di un ciclo quadruplo: 80-210, 90-200, 110-160, 120-150. La linea 130 realizza la gestione del nuovo stato secondo le regole.
La visualizzazione di questo nuovo stato è ottenuta riutilizzando il doppio ciclo di visualizzazione F (230-300) e G (240-290).
La linea 340 permette di ricominciare.

PER GIOCARE: Nella versione automatica, basta lanciare il programma con RUN e guardare.
Per fermarsi, bisognerà battere BREAK.
Nella versione manuale, la configurazione di inizio sarà inserita, dopo il lancio del programma con RUN, battendo 0 se non si vuole mettere niente sulla linea e sulla colonna indicate, e 1 se si desidera metterci una cellula. Poi l'evoluzione si svolgerà in seguito automaticamente.

ESTENSIONI POSSIBILI: Sarà facile, se si dispone di una memoria più capace, di aumentare le dimensioni degli arrays B e C, ridefinendo la variabile A. Si potrà anche introdurre il colore, migliorare la precisione del disegno rimpiazzando i PRINT con una visualizzazione ad alta risoluzione.

```
5 REM GIOCO DELLA VITA (MANUALE)
10 DIM N$(5),R$(3)
15 GRAPHICS 0
20 SETCOLOR 1,4,14:SETCOLOR 2,10,6
25 PRINT "DIMENSIONE :";:GOSUB 400:A=R:IF A>10 OR A<1 THEN 15
30 DIM B(A,A),C(A,A)
35 FOR I=1 TO A
40 FOR J=1 TO A
45 B(I,J)=0:C(I,J)=0
50 NEXT J
55 NEXT I
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 PRINT "}"
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 280
260 POSITION 0,22:PRINT "Riga ";F;" Colonna ";G;"-Battere 1 o 0:        ";
270 POSITION 33,22:GOSUB 400:C(F,G)=R:IF C(F,G)<>0 AND C(F,G)<>1 THEN 260
280 POSITION 2*G,2*F:IF C(F,G)=1 THEN PRINT "*"
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 N=N+1
330 N$=STR$(N)
340 POSITION 12-INT(LEN(N$)/2),23:PRINT "GENERAZIONE ";
350 FOR I=1 TO LEN(N$)
360 PRINT CHR$(ASC(N$(I,I))+128);
370 NEXT I
380 GOTO 80
400 INPUT R$:R=0
410 IF R$="" THEN R=0:RETURN
420 FOR I=LEN(R$) TO 1 STEP -1
430 AS=ASC(R$(I,I))-48:IF AS<0 OR AS>9 THEN R=0:RETURN
440 R=R+AS*10^(LEN(R$)-I)
450 NEXT I
460 RETURN
```

```
10 REM GIOCO DELLA VITA (AUTOMATICO)
15 GRAPHICS 0
20 SETCOLOR 1,10,6:SETCOLOR 2,4,14
25 PRINT "DIMENSIONE :";:INPUT A:IF A>11 OR A<0 THEN 15
30 DIM B(A,A),C(A,A),N$(5)
35 FOR I=1 TO A
40 FOR J=1 TO A
45 B(I,J)=0:C(I,J)=0
50 NEXT J
55 NEXT I
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 PRINT "}"
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 IF RND(1)<0.5 THEN C(F,G)=1
270 POSITION 2*G,2*F:IF C(F,G)=1 THEN PRINT "*"
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 N=N+1
315 N$=STR$(N)
320 POSITION 12-INT(LEN(N$)/2),23:PRINT "GENERAZIONE ";
325 FOR I=1 TO LEN(N$)
330 PRINT CHR$(ASC(N$(I,I))+128);
335 NEXT I
340 GOTO 80
```

LIVELLO 3

N.	Nome del programma	Tema e particolarità
1	Un dado	Lancio di un dado, con grafica
2	Più grande, più piccolo	Un gioco di deduzione interessante, per 1 giocatore
3	Ordine	Un po' di astuzia per rimettere in ordine, il più rapidamente possibile
4	L'impiccato	Un gioco di lettere molto classico (2 giocatori)
5	Briscola	Per simulare una mano, con un mazzo di 32 carte
6	Taccuino di calcolo	Un buon allenamento (2 giocatori)
7	Impiccato inverso	Questa volta è il calcolatore che cerca d'indovinare la parola
8	Nicomat	Per perfezionarsi in calcolo mentale (1 o più giocatori)
9	Corsa di cavalli	Abbastanza visuale: un gioco di fortuna, per vari educatori
10	Attraverso	Un gioco di strategia per 2 giocatori
11	Mastermind	Con dei numeri, ma classico, per 1 solo giocatore
12	Idem	Un gioco di calcolo mentale, con vari livelli, per vari giocatori
13	Orologio	Non buttate il vostro vecchio televisore; trasformatelo in un orologio
14	Regine sulla scacchiera	Un vecchio problema, per aiutarvi a riflettere
15	Ripetizioni	Ripetere una sequenza di cifre sempre più lunga
16	Parole	Un'idea molto semplice, ma non è così facile da trovare
17	Permutazioni	Un gioco di riflessione, non sempre evidente
18	Bridge	Mettete a punto i vostri problemi di bridge
19	Hanoi	Un consiglio: cominciate da poco!!!
20	Rombi	Per aiutarvi ad addormentarvi
21	Matteo	Non sapete disegnare? Lasciatelo fare a un calcolatore
22	Il lupo e gli agnelli	Un gioco di riflessione, per 2 giocatori

Manipolazione di stringhe e grafici ad alta risoluzione

TRATTAMENTO DI STRINGHE DI CARATTERI

Abbiamo avuto l'occasione di incontrare, già dal livello 2, la funzione essenziale utilizzata nel Basic ATARI per manipolare le stringhe: si tratta di A$.(I,J) che permette di estrarre, nella stringa A$, una sottostringa di lunghezza J—I+1 a iniziare dal carattere I.
Così se,

A$= "SIGNORE"

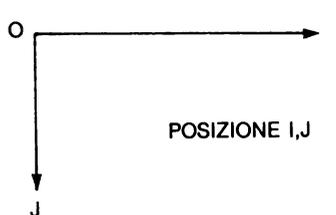
avremo:

A$.(2,4)= "IGN"

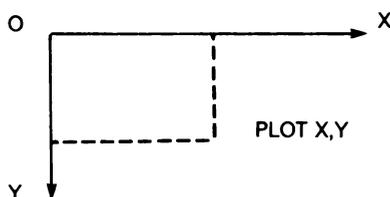
Il Basic dell'ATARI permette la sostituzione diretta di una sottostringa con una assegnazione del tipo A$ (—,—)= "—". Sarà necessario scrivere, per sostituire la Iesima lettera della stringa A$ (I,I)= "Nuova Lettera" e per allungare di 1 la stringa basterà fare: A$ (LEN (A$) + 1) = stringa di caratteri da aggiungere a destra.

LE DIFFERENTI FUNZIONI GRAFICHE

Fino ad ora abbiamo implicitamente utilizzato il comando GRAPHICS 0 per riportare lo schermo pulito e con i colori di base. Esistono due altre funzioni GRAPHICS 1 e GRAPHICS 2 grazie alle quali i caratteri battuti sono allargati. Così si ha:

per la funzione 0 --- 24 linee di 40 caratteri
per la funzione 1 --- 20 linee di 20 caratteri + 4 linee di testo normale
per la funzione 2 --- 10 linee di 20 caratteri + 4 linee di testo normale

La funzione POSITION è utilizzata in questi nuovi limiti, per posizionare la stampa che si ottiene con il comando PRINT @ 6 ; seguito dalla variabile (o dalla stringa di caratteri) desiderata.
Le funzioni grafiche da 3 a 15 permettono il disegno con una risoluzione più o meno dettagliata e con colori; così per le funzioni da 3 a 8:

	definizione (X,Y)	colori
funzione 3	40 x 20	4
» 4	80 x 40	2
» 5	80 x 40	4
» 6	160 x 80	2
» 7	160 x 80	4
» 8	320 x 160	1

La visualizzazione di un punto si ottiene fornendo un colore con il comando COLOR e posizionandolo con PLOT X,Y
Così per creare un punto in mezzo allo schermo con la funzione GRAPHICS 6, basterà fare: PLOT 80,40

Il colore delle visualizzazioni è determinato da SETCOLOR R, X, X.
L'ATARI dispone di 5 registri R (da 0 a 4) che contengono la gamma dei colori utilizzati per le scritte o per i disegni.

Il registro R = 0 fornisce il colore unicamente con funzione grafica
1 fornisce il colore unicamente del testo con funzione testo
2 fornisce il colore unicamente del fondo con funzione testo
3 fornisce il colore unicamente con funzione grafica
4 fornisce il colore dello schermo

Esempio: per avere un fondo rosso brillante, basta fare:

SETCOLOR 2,3,10

Il primo parametro 2 è il numero del registro che corrisponde al fondo.
Il secondo parametro 3 è il numero del colore.
Il terzo parametro 10 è il fattore luminoso.

N.B. aggiungendo 16 al numero di funzione grafica, si lavora su tutto lo schermo, recuperando le 4 linee di testo che rimangono.

Così, con il comando GRAPHICS 2 + 16 otteniamo dei caratteri allargati sulla totalità dello schermo.

Un dado

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di un dado (con grafica).
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma propone semplicemente il lancio di un dado a caso e la sua rappresentazione grafica in grande formato.
Potrà essere facilmente utilizzato come sottoprogramma in ogni tipo di gioco che faccia richiamo al lancio di uno o più dadi.

IL PROGRAMMA: La linea 50 utilizza il generatore casuale dell'ATARI per estrarre un numero da 1 a 6.
Il ciclo J (70-90) disegna i bordi del dado, le linee 60 e 100 disegnano l'alto e il basso
Le linee 120-170 vengono utilizzate per il disegno dei punti.
Le linee 180 e 190 permettono all'occorrenza di fare nuovamente il lancio.

PER GIOCARE: Lanciare il programma con RUN.
Si inserisce un intero per inizializzare il generatore, e si visualizza un dado colorato di grande formato.
Alla domanda ALTRO LANCIO (S/N) si risponde S per rilanciare il dado.

ESTENSIONI POSSIBILI: Si potrà introdurre una piccola musica che renderà più gradevole ogni lancio, fare sfilare i punti come se il dado rotolasse su una tavola, ecc.

```
10 REM UN DADO
20 DIM R$(1)
30 SETCOLOR 1,7,2:SETCOLOR 2,13,14:SETCOLOR 4,13,14
40 PRINT "}"
50 A=INT(RND(1)*6+1)
60 POSITION 12,5:PRINT "qrrrrrrrrrre":REM battere le minuscole schiacciando sim
ultaneamente <CONTROL>
70 FOR J=0 TO 10
80 POSITION 12,6+J:PRINT CHR$(124):POSITION 24,6+J:PRINT CHR$(124)
90 NEXT J
100 POSITION 12,17:PRINT "zrrrrrrrrrrc":REM vedere 60
120 IF A<>1 THEN POSITION 15,8:PRINT CHR$(20)
130 IF A>=4 THEN POSITION 21,8:PRINT CHR$(20)
140 IF A=6 THEN POSITION 15,11:PRINT CHR$(20):POSITION 21,11:PRINT CHR$(20)
150 IF A/2<>INT(A/2) THEN POSITION 18,11:PRINT CHR$(20)
160 IF A>=4 THEN POSITION 15,14:PRINT CHR$(20)
170 IF A<>1 THEN POSITION 21,14:PRINT CHR$(20)
180 POSITION 0,20:PRINT "Un'altro lancio (S/N)";:INPUT R$
190 IF R$="S" THEN 40
```

Più grande, più piccolo

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di deduzione (un solo giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Si ispira a un principio leggermente differente da quello del Mastermind; si tratta ugualmente di indovinare un numero (da 1 a 7 cifre), ma le indicazioni di cui si dispone sono questa volta da una parte il numero di cifre che, nel numero provato, sono superiori a quelle del numero da indovinare, e, dall'altra parte, il numero di cifre che sono inferiori. Se, per esempio, il numero da indovinare è 3054 e che si provi 4251, la macchina risponderà 2 più grandi (sono il 4 e il 2), e 1 più piccolo (è l'1).

IL PROGRAMMA: La linea 60 proibisce di chiedere dei numeri di più di 8 cifre.
Il numero da indovinare è posizionato, alla linea 80, in una variabile alfanumerica X$.
I paragoni col numero tentato sono fatti alla linea 130 (che testa se si è indovinato), e nel ciclo I (150-180), dove si fa il paragone "lettera per lettera".
La visualizzazione del risultato è realizzata grazie alla linea 190.
Il contatore del numero di tentativi è alla linea 200.

PER GIOCARE: Il lancio del programma con RUN provoca il messaggio NUMERO DI CIFRE.
Si inserisce un numero inferiore a 8 e si batte RETURN.
Il numero di tentativi appare così come la domanda NUMERO TENTATO; si risponde inserendo un numero, seguito da RETURN.
Il risultato è allora visualizzato nella maniera > e <.
Se non è esatto, si ricomincia (attenzione ad inserire unicamente dei numeri aventi tante cifre quante il numero da indovinare).
Se si indovina, l'ATARI scriverà AVETE TROVATO ... IN ... PROVE.
Poi UN'ALTRA PARTITA? (S/N); faremo S per cominciare una nuova partita.

ESTENSIONI POSSIBILI: Potremmo, come per il Mastermind, immaginare un gioco analogo con delle lettere, dei colori, o anche utilizzando degli altri simbolismi.
Si può anche prevedere di far giocare l'ATARI che avrà allora il ruolo di decodificatore.

```
10 REM PIU' GRANDE, PIU' PICCOLO
20 DIM X$(8),P$(8)
40 GRAPHICS 0
50 PRINT "Numero di cifre :";:INPUT C
60 IF C>8 THEN 40
70 E=1:F=1
80 X$=STR$(INT(RND(1)*9^C+10^(C-E)))
100 POSITION 0,3:PRINT "  Prova No.  Numero tentato   >     <  "
110 POSITION 2,4:FOR P=1 TO 36:PRINT CHR$(18);:NEXT P
120 POSITION 6,F+4:PRINT E;"                  ":POSITION 19,F+4:INPUT P$
130 IF P$=X$ THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF P$(I,I)>X$(I,I) THEN A=A+1
170 IF P$(I,I)<X$(I,I) THEN B=B+1
180 NEXT I
190 POSITION 30,F+4:PRINT A;"     ";B
200 E=E+1:F=F+1:IF F=18 THEN F=1
210 GOTO 120
220 PRINT :PRINT "Trovato ";X$;" in ";E;" prove."
230 PRINT :PRINT "Un'altra partita (S/N) ";:INPUT X$
240 IF X$(1,1)="S" THEN 40
```

Ordine

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: media.

IL GIOCO: Le 10 prime lettere dell'alfabeto appaiono in disordine; bisogna riuscire a rimetterle in ordine, col minor numero possibile di tentativi, con delle permutazioni successive di due delle lettere; ma questo ha anche come inconveniente di far permutare attorno a lei le due lettere che la circondano, secondo il meccanismo spiegato qui sotto:

```
B A D G H I J C F E
0 1 2 3 4 5 6 7 8 9     : permutazione 83

B A D F H I J E G C
0 1 2 3 4 5 6 7 8 9     : risultato
```

IL PROGRAMMA: I cicli chiusi I (60-120) e J (80-100) definiscono una variabile alfanumerica A$, contenente le 10 prime lettere dell'alfabeto in disordine.
La linea 140 visualizza A$, e la linea 150 visualizza i numeri delle lettere da utilizzare per la permutazione.
Si inserisce la permutazione alla linea 190.
Le linee 200 e 210 trasformano la variabile alfanumerica inserita in una variabile numerica.
La linea 220 chiama una prima volta il sottoprogramma d'inversione.
La linea 250 chiama una seconda volta il sottoprogramma d'inversione.
La linea 260 contabilizza i tentativi.
La linea 180 controlla i risultati.

PER GIOCARE: Lanciare il programma con RUN.
La parola iniziale appare, ogni lettera essendo riferita ad un numero da 1 a 9. Alla domanda INVERSIONE (I/J), si risponde inserendo un numero di due cifre xy; x è il numero della prima lettera e y il numero della seconda. Le due inversioni sono allora effettuate in successione e si ricomincia.
Il numero di tentativi apparirà alla fine della partita.

ESTENSIONI POSSIBILI: Si può immaginare un gioco contro il calcolatore, che valuti il punteggio del giocatore in funzione delle sue capacità; si possono anche modificare le regole che gestiscono le permutazioni.

```
10 REM ORDINE
20 DIM A$(10),B$(10),N$(1),M$(1),R$(2)
40 GRAPHICS 0
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND(1)*10+65))
75 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF R$=A$(J,J) THEN 70
100 NEXT J
110 A$(LEN(A$)+1)=R$
120 NEXT I
130 S=1
140 POSITION 2,4:PRINT "Risultato :";A$
150 PRINT "            0123456789"
160 POSITION 2,9:PRINT "Prova numero ";S
170 POSITION 33,12:PRINT "         "
180 IF A$="ABCDEFGHIJ" THEN PRINT "Riuscito in ";S;" mosse.":END
190 POSITION 0,12:PRINT "Inversione (I e J con forma IJ) ";:INPUT R$
200 I=ASC(R$(1,1))-47
210 J=ASC(R$(2,2))-47
220 GOSUB 500
230 IF R$(1,1)="0" OR R$(1,1)="9" THEN 250
240 I=I-1:J=I+2
250 GOSUB 500
260 S=S+1
270 GOTO 140
500 B$=""
510 FOR T=1 TO LEN(A$)
520 IF T=I THEN N$=A$(T,T)
530 IF T=J THEN M$=A$(T,T)
540 NEXT T
550 FOR T=1 TO LEN(A$)
560 IF T=I THEN B$(LEN(B$)+1)=M$:GOTO 590
570 IF T=J THEN B$(LEN(B$)+1)=N$:GOTO 590
580 B$(LEN(B$)+1)=A$(T,T)
590 NEXT T
600 A$=B$
610 RETURN
```

L'impiccato

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: gioco di lettere (versione a 2 giocatori).
Difficoltà del gioco: media.

IL GIOCO: Sarebbe senz'altro difficile ritrovare il nome dell'inventore di questo gioco che esiste da tempo immemorabile, qualche volta praticato sotto forme assai differenti (per esempio, ricostituire un maialino lanciando dei dadi).
L'apparizione dei giochi su micro-calcolatore e la magia degli schermi video gli hanno fatto conoscere un rinnovo sicuro: in questa versione, uno dei giocatori deve inserire una parola di lunghezza qualunque. Il calcolatore visualizzerà allora dei puntini nella posizione delle lettere di quella parola e chiede al secondo giocatore di proporre una lettera. Ogni risposta corretta fa apparire la lettera nella sua posizione esatta nella parola, ed ogni risposta sbagliata fa progredire il disegno del patibolo.
Per vincere, basta riuscire a trovare la parola sconosciuta prima di essere impiccato.

IL PROGRAMMA: La parola da indovinare è introdotta all'inizio del programma (linea 30); 2 linee di DATA (2000-2010) contengono tutti gli elementi del disegno sotto forma di elementi grafici.
Il ciclo I (60-80) fabbrica una parola P$ che contiene i puntini nella posizione di ognuna delle lettere della parola iniziale.
Il ciclo I e J (120-140) si occupano del disegno due linee per due (più o meno completo, secondo il valore di P).
Le linee 150-160 visualizzano il risultato e arrestano il gioco.
La lettera provata è inserita alla linea 170 e testata in un ciclo I (190-210); il sottoprogramma 300-320 rimpiazza ogni puntino con la lettera corrispondente.

PER GIOCARE: Dopo il lancio del programma con RUN, uno dei giocatori deve inserire la sua parola e battere RETURN.
Dei puntini appaiono nella posizione di ognuna delle lettere della parola; e la domanda LA TUA LETTERA appare sullo schermo; l'altro giocatore risponde allora proponendo una lettera seguita da RETURN.
Se la lettera è giusta, rimpiazzerà uno o più puntini; altrimenti, il patibolo comincerà a disegnarsi.
La parola esatta sarà visualizzata nel caso in cui si perda.

```
10 REM L'IMPICCATO
15 DIM D$(7),P$(20),R$(20),S$(1),B$(600)
20 GRAPHICS 0
25 SETCOLOR 1,7,2:SETCOLOR 2,13,14:SETCOLOR 4,13,14
30 PRINT "La tua parola :";:INPUT R$
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$(LEN(P$)+1)="-"
80 NEXT I
90 P=0
95 PRINT "}"
100 POSITION 2,3:PRINT P$
110 IF P=0 THEN 150
115 RESTORE
120 FOR I=1 TO P
125 FOR J=1 TO 2
130 POSITION 22,2*I+J:READ D$:PRINT D$
135 NEXT J
140 NEXT I
150 IF P$=R$ THEN PRINT "BRAVO!":END
160 IF P=7 THEN PRINT "PERSO! La parola era ";R$:END
170 POSITION 2,20:PRINT "La tua lettera :";:INPUT S$
180 T=0
190 FOR I=1 TO LEN(R$)
200 IF R$(I,I)=S$ THEN GOSUB 300
210 NEXT I
220 IF T<>0 THEN 100
230 P=P+1
240 GOTO 100
300 T=1
310 P$(I,I)=S$
320 RETURN
2000 DATA ,  qre ,  srs,  ! !,  zwc,qrrsrre,!  !  !,!  !  !,  f a, f   a
2010 DATA b     v,b     v,h     j,        ,
3000 REM per i DATA battere simultaneamente la lettera minuscola e il tasto <CON
TROL>
3010 REM nei DATA battere <SHIFT>= al posto del punto esclamativo per ottenere u
n tratto verticale pieno
```

Briscola

RIFERIMENTI: Programmazione: media.
Categoria del gioco: gioco con le carte.
Difficoltà del gioco: bassa.

IL GIOCO: Il gioco proposto è piuttosto una simulazione, realizzata con 32 carte, della distribuzione di 4 mazzetti di 5 carte. Potrà essere utilizzato per lo studio di problemi facenti intervenire delle distribuzioni di 20 carte su 32.

IL PROGRAMMA: Alla linea 50, si crea un array che può contenere le 20 carte.
Il ciclo I (80-140) assicura la presa casuale di 20 numeri, presi da 32; il ciclo J (110-130) verifica che non si realizzi due volte la stessa presa.
Il secondo ciclo I (160-310) assicura in seguito la visualizzazione, i nomi delle carte vengono scritti in chiaro e il colore viene decodificato e visualizzato in chiaro) grazie alle linee da 210 a 280.

PER GIOCARE: Basta battere RUN per ottenere immediatamente l'apparizione di 4 mazzetti di 5 carte ognuno (una mano).
Un nuovo RUN permette di ottenere una nuova distribuzione.

ESTENSIONI POSSIBILI: Si potrà facilmente mettere a punto una grafica migliore ridefinendo i caratteri dell'ATARI; ci si potrà ugualmente servire di questo programma come di un nucleo per la messa a punto di programmi di giochi di carte su calcolatore (che potrà, se necessario, servire da partner: vedere l'esempio del Poker, in questo stesso libro).

```
10 REM BRISCOLA
20 GRAPHICS 0
30 SETCOLOR 1,12,2:SETCOLOR 2,12,14
40 PRINT "}"
50 DIM A(20),S$(2),R$(4)
60 K=1
70 POSITION 5,11:PRINT "PAZIENTATE! Mescolo le carte.";
80 FOR I=1 TO 20
90 A(I)=1+INT(RND(1)*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF A(I)=A(J) THEN 90
130 NEXT J
140 NEXT I
150 PRINT "}"
160 FOR I=1 TO 20
170 S$="QU"
180 S=1+A(I)-8*INT(A(I)/8)
190 R$=STR$(S+6):R$(LEN(R$)+1)="  "
200 R$=R$(1,2)
210 IF S=5 THEN R$="FA"
220 IF S=6 THEN R$="DO"
230 IF S=7 THEN R$="RE"
240 IF S=8 THEN R$="AS"
250 IF A(I)<=24 THEN S$="FI"
260 IF A(I)<=16 THEN S$="PI"
270 IF A(I)<=8 THEN S$="CU"
280 POSITION 2+10*INT((I-1)/5),2*K:PRINT R$;" ";S$
290 K=K+1
300 IF K=6 THEN K=1
310 NEXT I
```

Taccuino di calcolo

RIFERIMENTI: Programmazione: media.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: elevata.

IL GIOCO: Si pratica con due giocatori, uno prende il ruolo di codificatore, l'altro quello di decodificatore.
Il codificatore parte da un numero (proposto dall'ATARI) e deve trasformarlo con l'aiuto di 5 operazioni che sceglie lui stesso, così come gli operatori, d'altronde.
Il risultato che ha ottenuto e le operazioni che ha utilizzato saranno visualizzate sullo schermo; il decodificatore dovrà allora cercare di ritrovare il numero iniziale effettuando le operazioni in senso inverso.

IL PROGRAMMA: La linea 50 permette di ottenere l'intero di partenza compreso tra 5 e 20.
Un ciclo I (110-200) permette di inserire successivamente le 5 operazioni salvaguardate nella stringa unica R$.
3 sottoprogrammi sono utilizzati per i calcoli (alle linee 1000 e 3000).
Dopo ogni calcolo, il risultato è visualizzato (solo il codificatore dovrà guardare lo schermo durante questa operazione) e il risultato definitivo resterà visualizzato per il decodificatore alla fine dei calcoli (linea 190).
Le linee 220-240 visualizzano le 5 operazioni utilizzate, così come gli operatori corrispondenti.

PER GIOCARE: Si batte RUN per lanciare il programma.
Appare il numero di partenza, poi la richiesta INSERITE 5 OPERAZIONI; si dovranno inserire le une dopo le altre nella maniera, per esempio, : +4, seguito da RETURN.
Il risultato sarà visualizzato dopo ogni operazione inserita. Alla fine, le operazioni inserite sono tutte visualizzate sulla stessa linea, ed il messaggio CORAGGIO è inviato all'attenzione del decodificatore (non esiste controllo nella decodifica).

ESTENSIONI POSSIBILI: Si potrà, naturalmente, pensare di migliorare la parte decodifica del programma introducendo dei controlli sui risultati; si può anche pensare di lasciar partecipare maggiormente l'ATARI affidandogli il ruolo della codifica, ecc.

```
10 REM TACCUINO DI CALCOLO
20 DIM R$(20),C$(8)
30 PRINT "}"
40 SETCOLOR 1,9,14:SETCOLOR 2,9,2
50 X=INT(RND(1)*16+5)
60 PRINT "Numero iniziale :";X
70 PRINT "Inserite 5 operazioni"
80 PRINT "Con la forma +5, /2, *4 ..."
90 F=X
110 FOR I=1 TO 5
120 POSITION 16,6:PRINT "              "
130 POSITION 2,6:PRINT "La tua operazione :";:INPUT C$
135 C$(LEN(C$)+1)="    ":R$(1+4*(I-1),4*I)=C$
140 A=ASC(C$(1,1))-41
150 IF A<=0 OR A=3 OR A=5 OR A>6 THEN 120
160 IF A>3 THEN A=(A/2)+1
170 ON A GOSUB 1000,2000,2000,3000
180 POSITION 16,10:PRINT "          "
190 POSITION 2,10:PRINT "Numero ottenuto :";F
200 NEXT I
210 PRINT "}":PRINT :PRINT :PRINT "Operazioni :";
220 FOR I=1 TO 5
230 PRINT R$(1+4*(I-1),4*I);" ";
240 NEXT I
250 PRINT :PRINT :PRINT "Coraggio...."
260 END
1000 F=F*VAL(C$(2,LEN(C$)))
1010 RETURN
2000 F=F+VAL(C$)
2010 RETURN
3000 F=F/VAL(C$(2,LEN(C$)))
3010 RETURN
```

Impiccato inverso

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: gioco sulle parole.
Difficoltà del gioco: bassa.

IL GIOCO: L'Impiccato è un gioco di lettere molto noto, già presentato d'altronde in questo libro, nel quale si deve indovinare una parola nascosta dal calcolatore e dove ogni lettera non scoperta vi porta un po' più vicini al patibolo.
Qui è l'ATARI che cercherà di indovinare la vostra parola, ma lui gioca un po' a caso e non possiede la vostra intuizione; ci vorranno dunque, in regola generale, molti tentativi per scoprire la soluzione. Per aumentare le sue possibilità, ci appelliamo a una formula che gli permette di pescare di più nelle vocali, poi nelle consonanti di uso comune e quindi nelle consonanti più rare.

IL PROGRAMMA: La linea 50 permette di registrare, in una variabile alfanumerica D$, le lettere dell'alfabeto, nell'ordine: vocali, consonanti usuali, altre consonanti. La parola da indovinare è introdotta alla linea 70. Il ciclo I (90-120) stampa dei puntini nelle posizioni delle lettere da trovare.
Il ciclo alla linea 140 serve unicamente a simulare il tempo di ricerca di una lettera.
La formula, piazzata alla linea 150, permette di modificare leggermente la probabilità di pesca, in favore delle vocali e delle consonanti usuali.
La linea 170 provoca una nuova pesca, e il ciclo I (210-240) ristampa i puntini e le lettere trovate.
La linea 250 incrementa il contatore dei tentativi.
La linea 270 visualizza il numero di tentativi realizzati, alla fine della partita.

PER GIOCARE: Lanciare il programma con RUN.
Inserire la parola da scoprire, seguita da RETURN.
I puntini sono allora visualizzati (uno per lettera) e l'ATARI propone una prima lettera, guarda se va bene e continua automaticamente, fino a che avrà trovato la parola. Si potrà uscire dal programma ad ogni momento, battendo BREAK.

ESTENSIONI POSSIBILI: Sarà senz'altro interessante, per i bambini di prevedere una versione non automatica dove il giocatore risponderebbe SI o NO per ogni lettera proposta dall'ATARI; si potrà anche immaginare di aggiungere una visualizzazione grafica, per esempio un patibolo o una macchina che va verso un burrone, ecc.

```
10 REM IMPICCATO INVERSO
30 DIM D$(26),E$(1),R$(30),T$(30)
40 PRINT "}"
50 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
60 C=1:T$=""
70 PRINT "Immetti la parola :";:INPUT R$
80 PRINT "}"
90 FOR I=1 TO LEN(R$)
100 POSITION 2+I,5:PRINT "-"
110 T$(LEN(T$)+1)="-"
120 NEXT I
130 POSITION 0,9:PRINT "Cerco....; prova No ";C;
140 FOR J=1 TO 100:NEXT J
150 X=6*(RND(1)<0.7)+13*(RND(1)<0.5)+7*(RND(1)<0.4)
160 X=INT(X*RND(1)+1)
170 IF D$(X,X)="-" THEN 150
180 E$=D$(X,X)
190 D$(X,X)="-"
200 POSITION 0,13:PRINT "Prendo :";E$;
210 FOR I=1 TO LEN(R$)
220 IF R$(I,I)=E$ THEN T$(I,I)=E$
230 POSITION 2+I,5:PRINT T$(I,I);
240 NEXT I
250 C=C+1
260 IF T$<>R$ THEN 130
270 POSITION 0,17:PRINT "TROVATO in ";C-1;" prove.";
280 FOR T=255 TO 0 STEP -1:SOUND 0,T,10,T-4*INT(T/4):NEXT T
290 SOUND 0,0,0,0
```

Nicomat

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: gioco di riflessione (calcolo mentale).
Difficoltà del gioco: media.

IL GIOCO: Utilizza una proprietà un po' particolare degli interi da 1 a 100 che possono essere identificati senza ambiguità dai loro resti nelle divisioni per 3, 5 e 7.
Il programma, proposto qui, è un programma codificatore che visualizza semplicemente i resti di un intero dato N nelle divisioni indicate.
Il gioco consiste nel cercare di scoprire l'intero (ma, in questa versione, avrete diritto ad un solo tentativo).

IL PROGRAMMA: La linea 50 permette di creare l'intero.
Il ciclo I (80-130) calcola i tre resti successivi, ed il ciclo chiuso J (100-120) permette di farli apparire in una forma assai evidente, nelle giuste locazioni.
Le linee 60 e 70 sono lì per indicare di quale divisione si tratta.
Il numero proposto dal giocatore è inserito alla linea 140; è visualizzato e testato alle linee 150 e 160 (messaggio di fine partita : BRAVO o PERSO).
Le linee 180-190 si occupano di gestire la prossima partita, secondo la risposta del giocatore.

PER GIOCARE: Si lancia il programma battendo RUN.
I tre resti saranno allora visualizzati sulla stessa linea, simbolizzati con dei cuori (l'assenza di cuori significa: resto 0).
Alla domanda LA TUA RISPOSTA , si risponderà con un intero tra 0 e 100, seguito da RETURN.
Si otterrà in risposta un messaggio PERSO o BRAVO, poi la domanda LA STESSA O UN'ALTRA PARTITA? (S/N). Per fare un nuovo tentativo, si risponderà S seguiti da RETURN.

ESTENSIONI POSSIBILI: Si può immaginare di realizzare il programma decodificatore (si danno i resti e l'ATARI dovrà darvi N) e cercare se la proprietà può essere estesa (effettuando altre divisioni, per esempio).

```
10 REM NICOMAT
20 GRAPHICS 0
30 DIM R$(1)
40 PRINT "}"
50 X=INT(RND(1)*100+1)
60 PRINT "    Resto della divisione per:"
70 POSITION 12,5:PRINT "3":POSITION 20,5:PRINT "5":POSITION 28,5:PRINT "7"
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1))
95 IF A=0 THEN 130
100 FOR J=1 TO A
110 POSITION J+8*I+3,7:PRINT "*"
120 NEXT J
130 NEXT I
140 POSITION 2,10:PRINT "La tua risposta :";:INPUT N
150 IF N=X THEN PRINT :PRINT "BRAVO!"
160 IF N<>X THEN PRINT :PRINT "PERSO!"
170 FOR T=120 TO 10:SOUND 0,T,10,4:NEXT T:SOUND 0,0,0,0
180 PRINT :PRINT "Un'altra partita (S/N) ";:INPUT R$
190 IF R$="S" THEN 40
```

Corsa di cavalli

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: fortuna (vari giocatori).
Difficoltà del gioco: nessuna.

IL GIOCO: Si inserirà un numero di cavalli compreso tra 1 e 9 e la corsa comincerà immediatamente.
Ci potranno essere dei pareggi, poiché si terrà conto del fatto che l'ATARI fa avanzare i cavalli gli uni dopo gli altri, cominciando col numero 1; la classifica sarà visualizzata man mano, secondo gli arrivi (e tenendo conto dei pareggi).

IL PROGRAMMA: Un primo ciclo non formale (80-173) visualizza i cavalli nella loro posizione di partenza.
Un secondo ciclo I (100-180), li farà avanzare; la posizione di ogni cavallo sarà individuata con una cifra o una lettera in una variabile alfanumerica A$.
La linea 120 farà avanzare i cavalli.
La linea 140 genera gli arrivi.
La linea 150 realizza le stampe.
Il sottoprogramma 240-280 fa avanzare il contatore di arrivi, e stampa i risultati.

PER GIOCARE: Si lancia il programma con RUN.
Alla domanda NUMERO DI CAVALLI?: si risponde con un intero da 1 a 9, seguito da RETURN.
La corsa si svolgerà in seguito automaticamente.

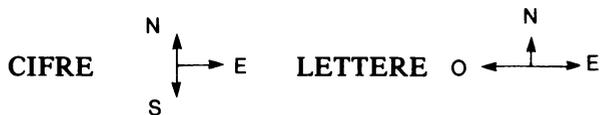
ESTENSIONI POSSIBILI: Potremmo disegnare i cavalli con una grafica più curata e differenziarli con dei numeri.
Potremmo anche immaginare un sistema di scommesse, che interverrà prima della partenza di ogni corsa.

```
10 REM CORSA DI CAVALLI
20 DIM A$(9),B$(1)
30 SETCOLOR 1,1,2:SETCOLOR 2,12,6
40 PRINT "}"
50 PRINT "Numero di cavalli :";:INPUT N:IF N>9 THEN 40
60 A$="":P=0:A=0:T=1:K=0
70 J=T
80 A$(LEN(A$)+1)="0":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF A$(I,I)="K" THEN 180
120 B$=CHR$(ASC(A$(I,I))+INT(RND(1)*4+1))
130 A$(I,I)=B$
140 IF B$>="K" THEN GOSUB 240
150 POSITION ASC(A$(I,I))-46,2*I+3:PRINT "     ";:PRINT I
160 SOUND 1,123,5,4*(I=3*INT(I/3))
170 IF K<>0 THEN 180
173 J=J+1:IF J<=N THEN 80
176 K=1:GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 IF A<N THEN 100
220 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR T=1 TO 20:NEXT T:NEXT I
230 POSITION 0,21:END
240 A$(I,I)="K"
250 POSITION 0,2*I+3:PRINT P
260 T=1
270 A=A+T
280 RETURN
1000 DATA 126,81,96
```

Attraverso

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO: Sullo schermo appare un quadrato di 9 caselle con lettere sul bordo verticale e cifre sul bordo orizzontale.
Uno dei giocatori manipola le cifre, l'altro le lettere.
Tre spostamenti sono possibili per le cifre, e altri 3 per le lettere. Le lettere possono spostarsi verso l'alto, a destra e a sinistra, e le cifre verso la destra, in alto o in basso.
Il vincitore è il primo che riesce a fare uscire tutti i suoi simboli (le lettere escono in alto, le cifre a destra).
Non ci sono prese, e i salti non sono autorizzati; non ci si può neanche posizionare su una casella già occupata.

CIFRE N ↑ → E ↓ S   LETTERE O ← ↑N → E

IL PROGRAMMA: Le posizioni delle cifre e delle lettere sono registrate in un array N (5,5); Le linee da 10 a 70 definiscono i valori di partenza, utilizzando gli indici 0 e 5 per i bordi.
Le linee da 90 a 140 definiscono il tavolo di gioco e le linee da 160 a 167 visualizzano il giocatore in gioco.
La linea 190-195 controllano il colpo giocato, così come i cicli I e J (200-240), ed il cambiamento di giocatore è assicurato dalla linea 330.

PER GIOCARE: Il tavolo di gioco apparirà dopo aver battuto RUN, e il calcolatore aspetta la mossa del giocatore individuato.
Si inserirà prima la cifra o la lettera rappresentante la pedina da spostare, seguita da una indicazione di direzione nella forma N (per Nord), S (per Sud), E (per Est), O (per Ovest).
Si potrà così fare 2E seguito da RETURN.
Se la mossa non è valida, non sarà eseguita, ed il codice del giocatore rimarrà alla visualizzazione; se è valida, lo spostamento sarà effettuato, e si otterrà il messaggio corrispondente al secondo giocatore.
Il primo che riuscirà a far sparire le sue 3 pedine avrà vinto.

ESTENSIONI POSSIBILI: Si può immaginare di estendere il gioco con più cifre e lettere e di migliorare i controlli. Si può anche definire una strategia per l'ATARI.

```
10 REM ATTRAVERSO
20 DIM N(5,5),R$(6):K=1
30 FOR I=0 TO 5
35 FOR J=0 TO 5
40 N(I,J)=0
45 IF J=1 THEN N(I,J)=ASC("0")+I
50 IF I=4 THEN N(I,J)=ASC("?")+J
55 IF I=0 OR I=5 OR J=0 OR I=5 THEN N(I,J)=1
60 NEXT J
65 NEXT I
70 N(4,1)=0
80 GRAPHICS 2
90 FOR U=1 TO 4
100 FOR V=1 TO 4
110 REM
120 POSITION 2*U,2*V-2:PRINT #6;CHR$(46*(N(V,U)=0)+N(V,U))
130 NEXT V
140 NEXT U
160 POSITION 1,9:PRINT #6;"LE ";:IF K=0 THEN PRINT #6;"LETTERE";
165 IF K<>0 THEN PRINT #6;"CIFRE";
167 PRINT #6;" GIOCANO";
180 PRINT "}":PRINT "La tua mossa (N,S,E,O)";:INPUT R$:R$(LEN(R$)+1)="     "
190 IF (K=1 AND R$(2,2)="O") OR (K=0 AND R$(2,2)="S") THEN 180
195 AS=ASC(R$(1,1)):IF (K=0)*(AS-52)<(K=1)*(AS-64) THEN 180
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=AS THEN 260
230 NEXT J
240 NEXT I
250 GOTO 180
260 A=I:B=J
270 I=I+(R$(2,2)="S")-(R$(2,2)="N")
280 J=J+(R$(2,2)="E")-(R$(2,2)="O")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) THEN 320
300 IF N(I,J)<>0 THEN 180
310 N(I,J)=ASC(R$)
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80
```

Mastermind

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: logica (un solo giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Diventato un grande classico dei giochi su personal computer, questo gioco sviluppa il ragionamento deduttivo; bisogna riuscire a trovare il più velocemente possibile un numero di n cifre (potremo prendere n compreso tra 1 e 6), disponendo ad ogni tentativo di informazioni.
Infatti, l'ATARI risponderà indicando da una parte il numero di cifre che sono, nel numero proposto, esattamente nella stessa posizione che nel numero da scoprire, e, dall'altra parte, il numero di cifre che figurano nel numero da scoprire senza essere esattamente nel posto giusto.

IL PROGRAMMA: La scelta del numero di cifre del numero da scoprire è effettuata alle linee 40 e 50.
Il numero è creato alla linea 60 (potranno esserci delle cifre ripetute).
La visualizzazione è preparata dalla linea 90. La linea 110 effettua la scelta pulendo per precauzione la linea visualizzata sullo schermo.
La linea 130 si occupa della fine della partita.
Un primo ciclo I (140-230) testa le cifre che sono nel posto giusto.
Un secondo ciclo J (160-260) testa le cifre che sono presenti e mal posizionate.
I contatori sono alle linee 200 e 220.
La linea 240 visualizza il risultato, e la linea 250 contabilizza i tentativi.

PER GIOCARE: Si batte RUN e si sceglie la lunghezza del numero desiderato (da 1 a 6), seguita da RETURN.
Dopo il messaggio N. NUMERO GIUSTO POSIZIONE G., si introduce il proprio tentativo, seguito da RETURN.
Si ottiene immediatamente la risposta (in caso di doppia cifra, i risultati possono essere un po' delicati da interpretare).
Si introduce in seguito un nuovo tentativo, e così via.
Quando si trova, si otterrà il messaggio: TROVATO IN . . . PROVE.
Si può uscire dal programma in ogni momento con BREAK.

ESTENSIONI POSSIBILI: Si può cercare di evitare la presenza di cifre doppie nel numero da trovare, si può anche prevedere un cambiamento automatico di pagina, un richiamo dei risultati precedenti; potremo anche realizzare dei giochi analoghi con delle parole, o altri simboli (colori, forme...)

```
10 REM MASTERMIND
20 DIM X$(6),E$(6)
40 PRINT "}":PRINT "Quante cifre :";:INPUT N
50 IF N>6 THEN 40
60 E=1:F=1:X$=STR$(INT(RND(1)*9^N+10^(N-1)))
80 PRINT "}"
90 PRINT " Nu.        NUMERO      GIUSTO  POSIZIONE"
100 FOR I=1 TO 37:PRINT CHR$(18);:NEXT I
110 POSITION 2,3+F:PRINT E;"                                    ";:POSITION 12,3+F
:INPUT E$
115 IF LEN(E$)<>N THEN 110
120 X=0:Y=X
130 IF E$=X$ THEN PRINT "TROVATO in ";E;" prove."
135 IF E$=X$ THEN FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR T=1 TO 15:NEXT T:NEXT
 I:END
140 FOR I=1 TO N
150 IF X$(I,I)=E$(I,I) THEN 220
160 FOR J=1 TO N
170 IF J<>I AND X$(I,I)=E$(J,J) THEN 200
180 NEXT J
190 GOTO 230
200 Y=Y+1
210 GOTO 230
220 X=X+1
230 NEXT I
240 POSITION 25,3+F:PRINT X;"          ";Y
250 E=E+1:F=F+1:IF F=18 THEN F=1
260 GOTO 110
500 DATA 126,81,96
```

Idem

RIFERIMENTI:

Programmazione: di media difficoltà.
Categoria del gioco: gioco di calcolo mentale (2 giocatori).
Difficoltà del gioco: secondo il livello scelto.

IL GIOCO:

L'ATARI sceglie un numero n di cifre (secondo il livello di difficoltà indicato all'inizio). Questo numero (lo stesso all'inizio per i 2 giocatori) sarà regolarmente decrementato, ad ogni giro, di una quantità ugualmente scelta a caso dall'ATARI all'inizio del gioco; voi dovete, ad ogni colpo, scegliere una colonna (le decine, le centinaia, ecc.) ed aggiungere in questa colonna un certo intero; lo scopo è di riuscire ad ottenere ad un certo momento la visualizzazione di un numero avente tutte le cifre identiche.
Ogni risultato sarà visualizzato un breve istante, dopo di che la mano passerà al giocatore seguente, che ripartirà dal suo risultato precedente.

IL PROGRAMMA:

Una variabile numerica N è dimensionata a 2 alla linea 40, per contenere i risultati successivi di due giocatori.
La linea 50 permetterà di inserire il livello di difficoltà scelto.
Il numero iniziale è creato dalla linea 70. La quantità che interverrà nel decremento è ottenuta alla linea 90.
La linea 120 testa se il risultato non è troppo piccolo.
La linea 130 visualizza il numero del giocatore.
Il numero è visualizzato alla linea 140 e il gioco del giocatore è inserito alla linea 160 (numero di due cifre).
La linea 170 incrementa la colonna scelta, e la linea 180 decrementa il numero della quantità scelta in partenza.
Il risultato sarà visualizzato un breve istante (linea 210). Il ciclo I (260-280) permette di verificare se tutte le cifre sono identiche.
La stampa e la fine della partita sono effettuate alla linea 290. Le linee 300 e 310 si occupano del cambiamento del giocatore.

PER GIOCARE:

Lanciare l'esecuzione del programma con RUN.
Il numero iniziale sarà visualizzato per il giocatore n. 1, che gioca battendo xy, seguito da RETURN; x è la cifra da aggiungere, e y è il numero della colonna dove bisogna aggiungerla.

Il risultato (provvisorio se è troppo piccolo) sarà allora visualizzato per un breve istante, e la linea passa al secondo giocatore, e così via. Si può fermare in ogni momento il programma battendo BREAK.

ESTENSIONI POSSIBILI: Si potrà estendere il gioco a più di 2 giocatori, migliorare la scelta del numero che interviene nella decrementazione, migliorare i diversi controlli, lasciare i differenti risultati visualizzati, ecc.

```
10 REM IDEM
20 DIM R$(2)
30 REM
40 PRINT "}":H=1:DIM N(2)
50 PRINT "Livello (2-7) :";:INPUT N
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND(1)+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND(1)+U^(N-1))
100 IF H=1 THEN P=P+1
110 PRINT "}"
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
130 POSITION 0,3:PRINT "Giocatore ";H;"   Prova ";P
140 POSITION 0,6:PRINT "Numero :":POSITION 16-N,6:PRINT N(H)
150 POSITION 9,7:PRINT "||||||| "
155 POSITION 9,8:PRINT "7654321 "
160 POSITION 0,10:PRINT "Il tuo gioco (N e C con forma NC) ";:INPUT R$
165 IF VAL(R$(2,2))>N THEN 160
170 N(H)=N(H)+VAL(R$(1,1))*U^(VAL(R$(2,2))-1)
180 N(H)=N(H)-A
190 IF N(H)>U^N THEN N(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 POSITION 0,13:PRINT "Risultato :";:PRINT N(H);
220 RESTORE :FOR I=1 TO 3:READ V:SOUND 0,V,10,4:FOR T=1 TO 15:NEXT T:NEXT I:SOUN
D 0,0,0,0
230 FOR J=1 TO 1200:NEXT J
240 R$=STR$(N(H))
260 FOR I=1 TO LEN(R$)-1
265 IF LEN(R$)=1 THEN 300
270 IF R$(I,I)<>R$(I+1,I+1) THEN 300
280 NEXT I
290 POSITION 0,16:PRINT "BRAVO! Giocatore ";H
295 FOR I=1 TO 3:READ V:SOUND 0,V,10,4:FOR T=1 TO 15:NEXT T:NEXT I:END
300 H=H+1
310 IF H>=3 THEN H=1
320 GOTO 100
1000 DATA 126,108,96,126,81,96
```

Orologio

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO: Trasformate il vostro televisore in orologio numerico a visualizzazione in grande formato, grazie a questo programma che vi permetterà di animare una vetrina, di fare un orologio di fantasia, e che potrà prestarsi a molti altri usi.

IL PROGRAMMA: La linea 30 vi permetterà di inserire l'ora sotto forma di un intero di 4 cifre (per esempio: 1024 significa ore 10 24 minuti).
La linea 35 cancella il cursore.
Le linee 80-170 realizzano il disegno, scalando la stampa man mano (con l'aiuto della variabile K, incrementata alla linea 160).
Le linee da 180 a 220 assicurano la progressione dell'ora.
Le linee da 230 a 290 effettuano la visualizzazione dei due punti lampeggianti per i secondi.

PER GIOCARE: Subito dopo il lancio con RUN, il programma si mette in attesa dell'ora; si batte e si attende il "top" per battere RETURN.
L'ora appare molto rapidamente e l'orologio è in marcia. Per fermarlo, bisognerà battere BREAK.
L'ora sarà inserita sotto forma di un intero avente imperativamente 4 cifre, per le ore 5 e 6 minuti, si dovrà battere 0506 seguito da RETURN.

ESTENSIONI POSSIBILI: Sarà interessante modificare il programma per fare una sveglia, con la suoneria; si può anche migliorare la grafica , introdurre il colore.

```
10 REM OROLOGIO
15 DIM A$(7)
20 GRAPHICS 0:SETCOLOR 1,3,2:SETCOLOR 2,8,10:SETCOLOR 4,8,10
30 PRINT "Ora (con forma XXXX) ";:INPUT A$
35 POKE 752,1
40 PRINT "}"
50 K=1
60 A=VAL(A$(K,K))
70 IF A<>1 AND A<>4 THEN POSITION 6*K+3,1:PRINT CHR$(160);CHR$(160);CHR$(160)
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7) THEN POSITION 6*K+3,J:PRINT CHR$(160)
100 IF A<>5 AND A<>6 THEN POSITION 6*K+5,J:PRINT CHR$(160)
110 IF A=0 OR A=2 OR A=6 OR A=8 THEN POSITION 6*K+3,J+4:PRINT CHR$(160)
120 IF A<>2 THEN POSITION 6*K+5,J+4:PRINT CHR$(160)
130 NEXT J
140 IF A>1 THEN POSITION 6*K+4,5:PRINT CHR$(160);CHR$(160)
150 IF A<>1 AND A<>4 AND A<>7 THEN POSITION 6*K+3,9:PRINT CHR$(160);CHR$(160);CH
R$(160)
160 K=K+1
170 IF K<5 THEN 60
180 A=1+VAL(A$)
190 IF A=2360 THEN A=0
200 A$="000":A$(4)=STR$(A)
210 A$=A$(LEN(A$)-3,LEN(A$))
220 IF A$(3,4)="60" THEN A$(1,2)=STR$(VAL(A$(1,2))+1):A$(3,4)="00"
230 H=1
240 FOR J=1 TO 128
250 FOR I=1 TO 24
260 POSITION 19,5:IF H<>1 THEN PRINT ":":GOTO 270
265 PRINT " "
270 NEXT I
280 H=ABS(H-1)
290 NEXT J
300 SOUND 0,193,10,4:FOR TE=1 TO 20:NEXT TE:GOTO 40
```

Regine sulla scacchiera

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: elevata.

IL GIOCO: Il gioco utilizza un algoritmo molto potente per risolvere un problema che ha dato filo da torcere a ricercatori, anche celebri come il matematico GAUSS, che non è riuscito a scoprire tutte le soluzioni. Si tratta di trovare tutte le maniere di disporre delle regine su una scacchiera, in modo tale che nessuna possa mangiarne un'altra. Il programma è molto elastico, visto che disponete anche della possibilità di scegliere la dimensione della scacchiera, e i risultati saranno visualizzati sia in forma grafica che numerica.

IL PROGRAMMA: La linea 30 vi permette di scegliere la dimensione della scacchiera.
Una variabile D sarà dimensionata di conseguenza, alla linea 40 e rimessa a zero.
Un ciclo di esplorazione principale I (50-110) contiene un sottociclo J (80-100) che realizza il controllo di non allineamento (linea 90).
Se, nel corso dell'esplorazione, si trova un allineamento, si torna indietro, utilizzando le linee da 230 a 270 e si ricomincia.
Il ciclo F (130-210) garantisce la stampa della scacchiera, ed il ciclo K (160-180), che potremo eventualmente sopprimere, assicura la stampa del risultato sotto forma di un codice numerico corrispondente alla posizione delle pedine in ogni colonna.
La linea 220 permette un arresto dopo l'apparizione di ognuna delle soluzioni.

PER GIOCARE: Si lancia il programma con RUN, poi si batte la dimensione della scacchiera desiderata, e si fa RETURN.
Basta aspettare con pazienza (più la scacchiera sarà grande, più l'attesa sarà lunga). La prima soluzione apparirà, poi il programma si ferma, e bisogna battere RETURN per farlo ripartire.
Per terminare l'esplorazione senza aver avuto tutte le soluzioni, si batterà BREAK.

ESTENSIONI POSSIBILI: Si può immaginare di trovare degli algoritmi per tutti i problemi analoghi (posizione delle torri, spostamento di un cavaliere, ecc.).

```
10 REM REGINE SULLA SCACCHIERA
20 GRAPHICS 0:SETCOLOR 1,4,10:SETCOLOR 2,8,2
30 PRINT "Formato della scacchiera :";:INPUT S
35 DIM R$(3)
40 DIM D(S):FOR I=1 TO S:D(I)=0:NEXT I
50 I=1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J) THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 PRINT "}"
130 FOR F=1 TO S
140 POSITION 2*F+1,0:PRINT F
150 POSITION 0,2*F:PRINT F
160 FOR K=1 TO S
170 POSITION 2*K+1,2*F:PRINT "."
180 NEXT K
190 POSITION 2*D(F)+1,2*F:PRINT "*"
200 POSITION F,20:R$=STR$(D(F)):PRINT R$(1,1)
210 NEXT F
220 SOUND 0,81,10,4:FOR T=1 TO 15:NEXT T:SOUND 0,0,0,0
225 POSITION 5,22:PRINT "Battere <RETURN> per continuare ";:INPUT R$
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240
```

Ripetizioni

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di memoria (uno o più giocatori).
Difficoltà del gioco: media.

IL GIOCO: Dei numeri appariranno uno ad uno.
Il giocatore dovrà ad ogni mano ripeterli tutti, dall'inizio (il che significa che ne avrà uno in più da battere ogni volta).
Dovrà cercare di resistere il più possibile. Il miglior punteggio è memorizzato dopo ogni partita, il che permetterà eventualmente di giocare in molti.

IL PROGRAMMA: La linea 30 chiama un sottoprogramma (linee da 1000 a 1060) che dà la regola del gioco.
La linea 80 crea le cifre, che sono in seguito immagazzinate in una variabile alfanumerica A$.
La linea 90 permette la stampa dell'ultimo numero a caso a caratteri ingranditi e la durata della stampa è determinata dalla linea 100.
La linea 120 aspetta la risposta del giocatore, che è paragonata alla risposta corretta alla linea 130.
La linea 160 marca i punti.
La linea 170 controlla il miglior punteggio.
I messaggi di fine partita sono effettuati dalla linea 200 alla linea 240.

PER GIOCARE: Il lancio del programma con RUN provoca la visualizzazione della regola del gioco, poi di una cifra che scompare. L'apparecchio pone allora la domanda LA TUA SEQUENZA?
Gli si risponde ridando la cifra, seguita da RETURN.
Se è giusta, si ha per un breve istante il messaggio ESATTO. ATTENZIONE, poi una nuova cifra appare, scompare, e ancora la domanda LA TUA SEQUENZA? Si ribattono allora le due cifre già ottenute, seguite da RETURN.
Si continua così (3 cifre, poi 4, 5 ecc.) fino a che si commetta un errore, il che provoca l'apparizione del punteggio (e del miglior punteggio ottenuto fino ad allora), seguito dalla domanda UN'ALTRA PARTITA? (S/N); si batte S per continuare.
Si può anche lasciare il programma in ogni istante, battendo BREAK prima dell'apparizione di una cifra.

ESTENSIONI POSSIBILI: Si potranno realizzare dei giochi analoghi e più o meno perfezionati rimpiazzando le cifre con delle lettere, o dei colori, o dei suoni...

```
10 REM RIPETIZIONI
20 DIM A$(100),R$(100),B$(13)
30 GOSUB 1000
40 M=0
50 A$=""
60 GRAPHICS 2
70 H=0
80 X=INT(RND(1)*10):A$(LEN(A$)+1)=STR$(X)
90 GRAPHICS 2:POSITION RND(1)*20,4:PRINT #6;X
100 FOR I=1 TO 200+RND(1)*300:NEXT I
110 PRINT "}"
120 PRINT "Immetti la tua sequenza :";:INPUT R$
130 IF R$<>A$ THEN 190
140 PRINT :FOR I=1 TO 3:RESTORE :READ V:SOUND 0,V,10,4:FOR T=1 TO 15:NEXT T:NEXT
 I:SOUND 0,0,0,0
145 PRINT "          ESATTO! attenzione !"
150 FOR I=1 TO 400:NEXT I
160 H=H+1
170 IF M<H THEN M=H
180 GOTO 80
190 PRINT "}"
200 PRIMT "Il tuo punteggio e' di     ";H
210 PRINT "Miglior punteggio attuale ";M
220 PRINT :PRINT "Un'altra partita (S/N) ";:INPUT R$
230 IF R$="N" THEN FOR I=1 TO 6:SOUND 0,126,10,4*(I/2=INT(I/2)):FOR T=1 TO 5:NEX
T T:NEXT I:END
240 GOTO 50
1000 PRINT "}":GRAPHICS 0
1005 R$="|RIPETIZIONI|":A$=CHR$(17):B$=CHR$(26)
1010 FOR T=1 TO LEN(R$)-2
1015 A$(LEN(A$)+1)=CHR$(18):B$(LEN(B$)+1)=CHR$(18)
1020 NEXT T
1025 A$(LEN(A$)+1)=CHR$(5):B$(LEN(B$)+1)=CHR$(3)
1030 POSITION (40-LEN(R$))/2,0:PRINT A$
1035 POSITION (40-LEN(R$))/2,1:PRINT R$
1040 POSITION (40-LEN(R$))/2,2:PRINT B$
1045 PRINT :PRINT "      Dovete ripetere tutta la serie"
1050 PRINT "dei numeri apparsi dall'inizio del "
1060 PRINT "gioco.Ce n'e' uno in piu' ogni volta"
1070 POSITION 4,22:PRINT "Battere <RETURN> per continuare";:INPUT R$
1080 RETURN
2000 DATA 126,108,96:REM DO RE MI
```

Parole

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di lettere (per 1 o vari giocatori).
Difficoltà del gioco: media.

IL GIOCO: È difficile immaginare un gioco più semplice nel principio, che si accontenta di visualizzare in disordine le lettere prese da una trentina di parole. Ma cercate di giocare e vedrete che non è così facile; la difficoltà varia evidentemente secondo la lista di parole che avrete introdotto precedentemente in DATA.

IL PROGRAMMA: La linea 40 permette di dimensionare la variabile Z(I) che conterrà la parola X se è stata scelta.
Il ciclo I (70) prende una parola tra le 30 scritte in DATA alle linee 240-280. La linea 70 sceglie una parola a caso.
Il ciclo (100-150) controlla se una lettera non è già stata pescata e riprende le lettere in disordine.
La linea 160 visualizza le lettere e la linea 170 aspetta la vostra risposta, che è testata alla linea 180.
Il numero dei tentativi è incrementato alla linea 190.
La fine della partita è controllata dalla linea 200, ed il punteggio è visualizzato alla linea 220.
Si noterà in 500 la routine per suonare note sciolte.

PER GIOCARE: Si lancia il programma con RUN. Una serie di lettere appare allora in disordine e bisogna cercare di ritrovare la parola corrispondente.
Si batte la parola supposta, poi RETURN, e apparirà una nuova serie di lettere se si risponde S alla domanda UN'ALTRA PROVA? (S/N).
Il punteggio ed il numero di tentativi appaiono alla fine della partita.

ESTENSIONI POSSIBILI: Si può pensare di limitare il tempo di apparizione della parola, o fare sfilare le lettere una dopo l'altra sullo schermo.
Si può anche aumentare la difficoltà inserendo dei JOKERS, giocare in molti, tenere il conto dei punti, ecc.

```
10 REM PAROLE
20 DIM Z(30),A$(30),R$(30),S$(30),V$(30):S=0:E=0
30 FOR I=1 TO 30:Z(I)=0:NEXT I
40 X=INT(RND(1)*30+1)
50 IF Z(X)=1 THEN 40
60 RESTORE 240
70 FOR I=1 TO X:READ R$:NEXT I
80 Z(X)=1:V$=R$
90 GRAPHICS 0:A$=""
100 FOR I=1 TO LEN(R$)
110 Z=INT(RND(1)*LEN(R$)+1)
120 IF R$(Z,Z)="-" THEN 110
130 A$(LEN(A$)+1)=R$(Z,Z)
140 R$(Z,Z)="-"
150 NEXT I
160 PRINT "Le lettere sono :";A$:RESTORE 1000:GOSUB 500
170 POSITION 2,6:PRINT "La tua soluzione :";:INPUT S$
180 IF S$=V$ THEN S=S+1
190 E=E+1
200 POSITION 2,12:PRINT "Un'altra prova (S/N) :";:INPUT R$
210 IF R$(1,1)="S" THEN 40
220 POSITION 2,18:PRINT "Il tuo puntessio :";S;" in ";E;" prove."
230 RESTORE 2000:GOSUB 500:END
240 DATA PROSCIUTTO,ENERGUMENO,AFFILATO,ERUDITO,INSIGNIFICANTE,REALISTA
250 DATA IMPRESSIONANTE,ALLETTANTE,EPICO,INTREPIDO,CODARDO,CALCOLATORE
260 DATA TRAVE,BANDERILLA,SENATO,ATTUALE,CONSACRATO,DIEDRO
270 DATA LIBRO,TERMINE,FIRMA,MALEODORANTE,AMANTE,ARBITRARIO
280 DATA CALORE,OPPORTUNO,BUCCIA,ALTRUISTA,SPAZZACAMINO,NOVIZIO
500 FOR I=1 TO 3
510 READ V:SOUND 0,V,10,4
520 FOR T=1 TO 10:NEXT T
530 SOUND 0,0,0,0:FOR T=1 TO 5:NEXT T
540 NEXT I
550 RETURN
1000 DATA 126,108,96
2000 DATA 126,126,126
```

Permutazioni

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessione.
Difficoltà del gioco: media.

IL GIOCO: L'ATARI vi chiede prima il numero di lettere che desiderate permutare (non più di 11); visualizzerà quindi il numero corrispondente di lettere prese all'inizio dell'alfabeto, prima in disordine (INIZIO) poi in ordine (FINE).
Dovrete riuscire a rimettere nell'ordine le lettere date all'inizio, utilizzando delle permutazioni sulle n prime lettere partendo da sinistra, col minor numero di tentativi possibile.

IL PROGRAMMA: La linea 40 permette di scegliere il numero di lettere, e la linea 50 controlla che non se ne chiedano più di 11.
La linea 80 realizza la mescola, prendendo a caso la nuova posizione di ogni lettera. Le lettere sono in seguito aggiunte man mano che sono pescate, in una variabile B$ (linea 100), eliminate in A$ e rimpiazzate in C$, nell'ordine giusto (linee 110-120).
La linea 140 controlla il numero di lettere di B$.
Le visualizzazioni sono effettuate dalle linee 160, 170 e 180.
Il rigiro è fatto, lettera per lettera, nel ciclo I (210-230); il risultato è piazzato in B$, alla linea 240; visualizzato dalla linea 250 e paragonato al risultato finale alla linea 260.
Il numero di tentativi sarà visualizzato alla fine della partita, grazie alla linea 270.

PER GIOCARE: Dopo aver battuto RUN, appare la domanda NUMERO DI LETTERE (DA 1 A 11); si risponde battendo N, seguita da RETURN. La visualizzazione seguente presenta l'ordine iniziale (INIZIO) e l'ordine da ottenere (FINE).
Si gioca indicando il numero di lettere da permutare partendo da sinistra, per esempio: su un messaggio iniziale di GEABDFC, se si batte 3 e RETURN si otterrà AEGBDFC.
Si continuerà fino ad ottenere l'ordine desiderato.
Il punteggio apparirà alla fine della partita così come la domanda UN'ALTRA PARTITA? (S/N); rispondere S se si desidera continuare.

ESTENSIONI POSSIBILI: Si potrà utilizzare lo stesso tipo di gioco per, ad esempio, ricostituire un disegno nascosto, dove le

diverse parti sarebbero identificate con delle lettere. Si possono anche realizzare delle permutazioni più complesse (per esempio, il fatto di permutare due lettere, permuta anche le due vicine alla lettera più a sinistra, ecc.)

```
10 REM PERMUTAZIONI
20 DIM A$(11),B$(11),BE$(11),C$(11)
30 GRAPHICS 2
40 PRINT "Numero di lettere (tra 1 e 11) ";:INPUT N
50 IF N>11 THEN 40
60 A$="ABCDEFGHIJKL":C$="                   ":B$=""
70 GRAPHICS 2
80 X=INT(RND(1)*N+1)
90 IF A$(X,X)=" " THEN 80
100 BE$=B$:B$=A$(X,X):B$(LEN(B$)+1)=BE$
110 C$(X,X)=A$(X,X)
120 A$(X,X)=" "
130 S=1
140 IF LEN(B$)<>N THEN 80
150 C$=C$(1,LEN(B$))
160 POSITION 0,2:PRINT #6;"INIZIO  :";B$
170 POSITION 0,4:PRINT #6;"FINE    :";C$
180 PRINT "}";:PRINT "Permutazione No. ";S:PRINT :PRINT "Quantita' :";:INPUT L:I
F L>N THEN 180
190 S=S+1
200 A$=""
210 FOR I=1 TO L
220 A$(LEN(A$)+1)=B$(L-I+1,L-I+1)
230 NEXT I
240 B$(1,L)=A$
250 GOSUB 500:POSITION 0,6:PRINT #6;"RISULTATO:";B$
260 IF B$<>C$ THEN 180
270 GOSUB 500:GOSUB 500:PRINT "}":PRINT "      RIUSCITO in ";S-1;" prove."
280 PRINT :PRINT "    Un'altra partita (S/N)";:INPUT A$
290 IF A$(1,1)="S" THEN 30
300 GRAPHICS 0:END
500 FOR I=255 TO 0 STEP -3
510 SOUND 0,I,10,4
520 NEXT I
530 RETURN
```

Bridge

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: variabile, secondo il problema posto.

IL GIOCO: Non troverete, in questo programma un compagno per i vostri bridges, ma, più esattamente, una simulazione per la visualizzazione di una mano; questa visualizzazione è realizzata gioco per gioco. Si può, partendo da questo, porsi ogni sorta di problemi: dichiarazioni, gioco della carta, ecc.

IL PROGRAMMA: Le linee 50 e 60 definiscono un mazzo di 52 carte.
La variabile B$, dimensionata alla linea 70, permetterà di visualizzare i simboli dei colori e anche di migliorare il disegno di certe carte (10, fante, donna, re e asso).
Il ciclo J (80-220) realizza la pesca di quattro mazzi di 13 carte.
Il ciclo I (90-190) realizza pesca e stampa. La carta è scelta tra le 52 disponibili, alla linea 110.
La linea 110 calcola il colore della carta pescata.
La linea 120 verifica che questa carta non sia già stata pescata.
Le linee 130 e 140 si occupano della stampa complementare nel caso si peschi un 10, un fante, una donna, un re o un asso.
La stampa è realizzata dalle linee 150, 160 e 170.
La linea 180 permette di eliminare le carte già pescate, e la linea 200 realizza un arresto tra ogni visualizzazione.

PER GIOCARE: Si lancia il programma con RUN.
Una prima serie di 13 carte appare verticalmente. Basterà battere un RETURN per ottenere la seconda serie di 13 carte. Si procederà nello stesso modo per la terza e la quarta serie.

ESTENSIONI POSSIBILI: Si può immaginare di raffigurare le 13 carte in una maniera più convenzionale.
Si può anche aggiungere la possibilità di fare delle dichiarazioni (e di paragonarle a quelle dell'ATARI, per esempio), o introdurre delle possibilità di gioco della carta, compreso anche l'ATARI.

```
10 REM BRIDGE
20 DIM B$(13),R$(52),S$(1)
40 GRAPHICS 0
50 R$="234567891FDRA"
60 FOR I=1 TO 2:R$(LEN(R$)+1)=R$:NEXT I
70 B$="FIQUCUPI0ADES"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=INT(RND(1)*LEN(R$)+1)
110 S=2*INT((X-1)/13)+1
120 IF R$(X,X)="0" THEN 100
130 T=X-13*INT(X/13)
140 T=T+13*(T=0)
150 POSITION 10*J-8,I:PRINT R$(X,X);
160 IF T>=9 THEN PRINT B$(T,T)
170 POSITION 10*J-4,I:PRINT B$(S,S+1)
180 R$(X,X)="0"
190 NEXT I
200 IF J<4 THEN POSITION 4,18:PRINT "Battere <RETURN> per continuare";:INPUT S$
210 POSITION 4,18:PRINT "
220 NEXT J
```

Hanoi

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: elevata.

IL GIOCO: È presentato, qui, in una versione per un solo giocatore. Si dispone di un certo numero di dischi (8 al massimo) e si deve trasferirli dalla torre numero 1 alla torre numero 3, rispettando una sola regola: in nessun momento un disco può essere ricoperto da un disco di diametro superiore.
Il numero di tentativi non è contabilizzato in questo programma, ma le mosse illecite saranno rifiutate.

IL PROGRAMMA: La variabile E$, alla linea 40, servirà a disegnare i dischi.
Il numero di dischi è inserito alla linea 50 e rifiutato se ci sono più di 8 dischi.
Il ciclo I (70-90) permette di fabbricare una variabile A(1) che contiene lo stato della pila sotto forma numerica (alla linea 30, tre variabili sono previste per conservare lo stato delle tre pile).
Le basi sono disegnate dalla linea 110 e le tre pile dai cicli chiusi I (130-180) e J (140-170).
Il controllo della mossa è effettuato dalle linee 210, 220, 235, 237, 260 e lo stato delle pile è modificato dalle linee 270 e 280.

PER GIOCARE: Dopo un RUN che lancia il programma, si inserisce il numero di dischi richiesti, seguito da RETURN. Si ottiene il disegno dello stato iniziale e, alla domanda TORRE DI PARTENZA, si risponde dando il numero della torre di partenza. Questo numero (1, 2 o 3) sarà rifiutato se la torre è vuota. Poi si batte RETURN e alla domanda TORRE D'ARRIVO, si risponde col numero della torre d'arrivo.
Se la mossa non è possibile, saremo obbligati a ricominciare da TORRE DI PARTENZA. Se la mossa è possibile, lo spostamento è realizzato e si ricomincia.

ESTENSIONI POSSIBILI: Si potranno disegnare dei dischi di diversi colori, introdurre un contatore di mosse e, soprattutto, definire un gioco per il calcolatore.
Si può anche cercare di complicare il gioco con due torri in più e due giocatori, ognuno avente dei dischi di colore diverso, per esempio.

```
10 REM HANOI
15 DIM A$(8),B$(9),C$(9),D$(9)
20 GRAPHICS 0:SETCOLOR 1,6,2:SETCOLOR 2,2,14
30 DIM A(8):A(1)=9:A(2)=9:A(3)=9
40 FOR I=1 TO 9:D$(I)=CHR$(160):NEXT I
50 PRINT "Numero di torri (meno di 8) :";:INPUT N
60 IF N>=8 THEN 50
70 FOR I=0 TO N-1
80 A$=STR$(A(1)):A$(LEN(A$)+1)=STR$(N-I):A(1)=VAL(A$)
90 NEXT I
100 PRINT "}"
110 POSITION 3,14:PRINT D$:POSITION 15,14:PRINT D$:POSITION 27,14:PRINT D$
120 POSITION 7,14:PRINT "1":POSITION 19,14:PRINT "2":POSITION 31,14:PRINT "3"
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=STR$(A(J))
160 IF LEN(B$)>I+1 THEN POSITION 12*J-9,13-I:PRINT D$(1,VAL(B$(I+2,I+2)))
170 NEXT J
180 NEXT I
200 POSITION 0,16:PRINT "Torre di partenza ";:POSITION 19,16:INPUT D
210 IF D>3 OR D<1 THEN 200
220 IF A(D)=9 THEN 200
230 POSITION 0,18:PRINT "Torre d'arrivo :     ";:POSITION 16,18:INPUT A
235 IF A>3 OR A<1 THEN 230
237 IF A=D THEN 200
240 B$=STR$(A(D))
250 C$=STR$(A(A))
260 IF B$(LEN(B$))>C$(LEN(C$)) THEN 200
270 A(A)=VAL(C$)*10+VAL(B$(LEN(B$)))
280 A(D)=VAL(B$(1,LEN(B$)-1))
290 GOTO 100
```

Rombi

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: passatempo.
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma genera progressivamente sullo schermo un rombo poi lo cancella progressivamente per poi ricominciare e via di seguito.
Potrete utilizzarlo come sonnifero se le pecore non vi bastano più.
Nella versione leggermente modificata che vi è proposta appariranno dei motivi casuali dopo la costruzione di ogni rombo.

IL PROGRAMMA: Un primo ciclo K (55-130) definisce le dimensioni dei tracciati successivi.
Il secondo ciclo I (60-120) traccia delle bande di punti, utilizzando le istruzioni PLOT (linee 80-110).
Se la variabile S (linea 70) è a 1, si avrà una cancellazione casuale (si utilizza PLOT, ma servendosi dei colori casuali per cancellare).
Nel secondo programma si otterrà una cancellazione casuale modificando la linea 70.

PER GIOCARE: Si lancia l'esecuzione del programma con RUN. Si inserisce se necessario un nucleo per il generatore (secondo programma) e si guarda.
Si può fermare in qualsiasi momento battendo BREAK.

ESTENSIONI POSSIBILI: Si potrà introdurre il suono, utilizzare l'istruzione DRAWTO che permetterà un tracciato molto più rapido, immaginare su dei principi analoghi diversi programmi (caleidoscopio, tappezzeria...).

```
10 REM ROMBI
20 GRAPHICS 7+16
40 S=0
50 FOR I=0 TO 4:SETCOLOR I,INT(RND(1)*16),2*(I+2):NEXT I
55 FOR K=0 TO 40
60 FOR I=0 TO K
65 COLOR 1
70 IF S<>0 THEN ON INT(RND(1)*5+1) GOTO 160,170,180,190,200
80 PLOT 80-K+I,40-I
90 PLOT 80-K+I,40+I
100 PLOT 80+K-I,40+I
110 PLOT 80+K-I,40-I
120 NEXT I
130 NEXT K
140 S= NOT S
150 GOTO 50
160 COLOR 0:PLOT 80-K+I,40-I
170 COLOR 1:PLOT 80-K+I,40+I
180 COLOR 2:PLOT 80+K-I,40+I
190 COLOR 3:PLOT 80+K-I,40-I
200 GOTO 120
```

```
10 REM ROMBI INTERI
20 GRAPHICS 7+16
40 S=0
50 CO=1+INT(RND(1)*3)
55 FOR K=0 TO 40
60 FOR I=0 TO K
65 COLOR CO
70 IF S<>0 THEN 160
80 PLOT 80-K+I,40-I
90 PLOT 80-K+1,40+I
100 PLOT 80+K-I,40+I
110 PLOT 80+K-I,40-I
120 NEXT I
130 NEXT K
140 S= NOT S
150 GOTO 50
160 COLOR 0:PLOT 80-K+I,40-I
170 PLOT 80-K+I,40+I
180 PLOT 80+K-I,40+I
190 PLOT 80+K-I,40-I
200 GOTO 120
```

Matteo

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: disegno casuale.
Difficoltà del gioco: nessuna.

IL GIOCO: Si utilizza la definizione di una curva in coordinate polari e l'istruzione PLOT per ottenere sullo schermo dei disegni casuali (del tipo fiore) più o meno eleganti.
I due parametri principali A e B possono essere, se necessario, inseriti alla mano (istruzione INPUT) per fare uno studio più sistematico di diverse curve possibili.

IL PROGRAMMA: I primi 3 parametri Z, K e T, introdotti alla linea 40, definiscono il raggio massimo e l'occupazione dello schermo.
Le variabili casuali A e B scelte tra 0 e 2, dalla linea 60, definiscono l'andamento della curva; potrà essere interessante introdurle manualmente (per esempio: 60 INPUT A,B).
Il ciclo H (70-120) realizza la traccia punto per punto.
Il raggio R è calcolato alla linea 80 e il passaggio in coordinate cartesiane è effettuato alle linee 90 e 100.
La visualizzazione dei punti è realizzata dalla linea 110.
La linea 150 suona la musichetta contenuta in A$
La linea 180 sceglie un altro colore del disegno e il programma ricomincia in 40.

PER GIOCARE: Si lancia il programma con RUN.
Il disegno sarà eseguito automaticamente.
Per terminare battere BREAK.

ESTENSIONI POSSIBILI: Si potranno utilizzare delle funzioni grafiche più definite, variare i colori.
Si potrà cercare di utilizzare delle altre funzioni per realizzare dei tracciati ancora più straordinari (per esempio con simulazione della terza dimensione).

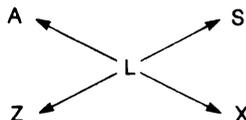
```
10 REM MATTEO
20 GRAPHICS 7+16:FOR I=0 TO 4:SETCOLOR I,INT(16*RND(1)),2*I:NEXT I
25 DIM A$(9)
30 C=INT(RND(1)*4):IF C=0 THEN 30
40 Z=5:K=9:T=3
60 A=3*RND(1):B=3*RND(1)
70 FOR H=0 TO 6.28 STEP 3.14/180
80 R=Z*SIN(H*T)
90 X=75+INT(K*R*COS(A*H))
100 Y=47+INT(K*R*SIN(B*H))
110 COLOR C:PLOT X,Y
120 NEXT H
140 A$="126096081"
150 FOR I=0 TO 2:SOUND 0,VAL(A$(3*I+1,3*I+3)),10,4:FOR T=1 TO 10:NEXT T:SOUND 0,
0,0,0:FOR T=1 TO 5:NEXT T:NEXT I
180 C=C+1:IF C=0 THEN 180
190 GOTO 40
```

Il lupo e gli agnelli

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO: È un gioco abbastanza conosciuto che può essere praticato su una scacchiera 8*8, o indifferentemente, su una 10*10. È presentato, qui, in una versione per una scacchiera 8*8; uno dei giocatori sceglie il lupo che si sposta seguendo le diagonali, restando sulle caselle nere, di una casella ad ogni colpo, e che può avanzare o retrocedere. Gli agnelli si spostano ugualmente sulle caselle nere, in diagonale, ma non hanno il diritto di retrocedere. Le caselle già occupate sono proibite (niente prese), e non ci sono salti.
Il lupo vince se riesce ad arrivare sulla linea in basso prima che tutti gli agnelli occupino quella in alto; gli agnelli vincono in questo caso, o se riescono a bloccare il lupo.

IL PROGRAMMA: Una variabile A$, alla linea 30, contiene le posizioni di partenza.
Il ciclo I (80-130) disegna la scacchiera, e le linee da 150 a 190 posizionano il lupo e gli agnelli.
La linea 200 gestisce il cambio di giocatore.
La linea 280 visualizza "TURNO DEL LUPO" se è il giocatore che manipola il lupo che deve giocare, e la linea 240 "TURNO DEGLI AGNELLI" nel caso contrario.
Le linee 250-270 o analizzano la risposta, in funzione del giocatore in questione.

PER GIOCARE: RUN posiziona il calcolatore in attesa di un inserimento: 0 e RETURN se si vuole far iniziare gli agnelli e 1 per il lupo.
Dopodiché appaiono la scacchiera e la frase TURNO DEL LUPO (A, S, Z, X) per il lupo e TURNO DEGLI AGNELLI (ND o NS) per gli agnelli.
Gli agnelli rispondono con un numero, seguito da D o S (per destra o sinistra). Così 2D provoca lo spostamento dell'agnello numero 2 che sale in diagonale di una casella a destra. È visualizzato lo spostamento effettuato, poi si ottiene una domanda identica per il lupo che risponderà battendo A, S, Z o X, secondo la direzione scelta:

Si dovrà fare attenzione a non uscire dalla scacchiera e a non finire sull'avversario, dato che non è previsto alcun controllo.

ESTENSIONI POSSIBILI: Si potrà prevedere una individuazione delle caselle, una grafica più curata, dei controlli di non superamento dei limiti del tavolo, di non arrivo su una casella già occupata...
Si potrà anche pensare di giocare su una scacchiera di 10 * 10 caselle, che aumenta leggermente la difficoltà del gioco per gli agnelli. Ma l'estensione più interessante da realizzare sarà certamente la creazione di un programma che permetta all'ATARI di comportarsi come un degno avversario , sia col lupo, sia con gli agnelli (e, in questo caso, sarà possibile definire una strategia vincente per l'ATARI).

```
10 REM IL LUPO E GLI AGNELLI
15 DIM A$(10),CA$(4),R$(2)
20 PRINT "Agnelli(0) o Lupo(1) :";:INPUT S
30 A$="8183858714"
35 FOR N=1 TO 4:CA$(N)=CHR$(160):NEXT N
40 GRAPHICS 0:SETCOLOR 1,10,14
50 PRINT "   A   B   C   D   E   F   G   H"
60 FOR I=1 TO 8:POSITION 0,2*I:PRINT I:NEXT I
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 POSITION 4*J-1,2*I
110 IF (I+J)/2=INT((I+J)/2) THEN PRINT CA$:POSITION 4*J-1,2*I+1:PRINT CA$:GOTO 1
20
115 PRINT "    "
120 NEXT J
130 NEXT I
150 POSITION 4*VAL(A$(10,10))+1,2*VAL(A$(9,9)):PRINT "L"
160 POSITION 4*VAL(A$(2,2))+1,2*VAL(A$(1,1)):PRINT "1"
170 POSITION 4*VAL(A$(4,4))+1,2*VAL(A$(3,3)):PRINT "2"
180 POSITION 4*VAL(A$(6,6))+1,2*VAL(A$(5,5)):PRINT "3"
190 POSITION 4*VAL(A$(8,8))+1,2*VAL(A$(7,7)):PRINT "4"
200 S=ABS(S-1)
210 POSITION 0,20:PRINT "                                       "
220 POSITION 0,20:IF S=0 THEN PRINT "Turno del lupo (A,Z,S,X) :";:INPUT R$
230 IF S=1 THEN PRINT "Turno degli agnelli (ND o NS) :";:INPUT R$:GOTO 280
240 C=10
250 A=VAL(A$(LEN(A$)-1,LEN(A$)))+(C+1)*(R$="X")+(C-1)*(R$="Z")
260 A=A-(C+1)*(R$="A")-(C-1)*(R$="S")
270 GOTO 300
280 C=2*VAL(R$(1,1))
290 A=VAL(A$(C-1,C))-9*(R$(2,2)="D")-11*(R$(2,2)="S")
300 A$(C-1,C)=STR$(A)
310 GOTO 80
```

LIVELLO 4

N.	Nome del programma	Tema e particolarità
1	Lancio di dadi	Sufficientemente elastico per essere usato dappertutto
2	Tombola	Vincete senza problemi alla prossima estrazione
3	La bomba	Saprete scoprirla in tempo? (1 giocatore)
4	Lavagna magica	Mettete a punto le vostre creazioni grafiche, senza carta né penna
5	Pubblicità	Se non sapete cosa mettere nella vostra vetrina
6	Cappelli	Siate molto osservatori e riuscirete (2 giocatori)
7	Combattimento aereo	Non è così facile abbattere un aereo che si sposta sempre
8	Disegno di dadi	Migliorate i vostri giochi di dadi inserendo questo programma
9	Il gioco del missile	Una missione pericolosa! (1 giocatore)
10	La scimmia	Un gioco di lettere per due giocatori, dove bisogna avere dei buoni riflessi
11	Il gioco del 15	Un gioco di riflessione che utilizza cifre (2 giocatori)
12	Destrezza	Gioco di riflessi per 1 giocatore
13	Sottomarino	Un classico dei videogiochi (1 giocatore)
14	Zombi	Un po' strano, ma molto bello (1 giocatore)
15	Diago	Siete veramente sicuri dei vostri riflessi?
16	Recomat	Per evitare di ammuffire su Nicomat
17	Caccia al sottomarino	Vi ci vorrà non poca riflessione e molta fortuna
18	Tastiera	Imparate a battere a macchina divertendovi

Le tecniche d'animazione elementare I caratteri grafici

PRESA AL VOLO DI UN CARATTERE BATTUTO ALLA TASTIERA:

Il principio di un gioco grafico, in animazione, è che si svolga da solo, in una certa maniera, durante un certo tempo, finché non interviene il giocatore.

Bisogna dunque disporre della possibilità di osservare se c'è stato o no l'intervento del giocatore, senza per questo interrompere lo svolgimento del programma.

Nel Basic dell'ATARI, questa proprietà è offerta dalla funzione GET che legge la tastiera interrompendo il programma. La sua sintassi molto particolare deve essere rispettata. Prima dell'utilizzo si inizializza con il comando:

OPEN # 1,4,0,"K"

che si pone all'inizio del programma. La scelta è effettuata da:

GET # 1, VAR

"VAR" contiene il codice ASCII del tasto battuto.

Per scegliere al volo è necessario utilizzare una particolarità dell'ATARI. Ad ogni tasto battuto corrisponde un codice speciale dell'ATARI (si parla di codice ATASCII) situato nella memoria 764. Basta leggere questa memoria per sapere quale tasto è stato premuto per ultimo.
Il comando:

PEEK legge il contenuto di una memoria:

così, CLA=PEEK(764)

assegna alla variabile CLA il codice tastiera dell'ultimo tasto battuto.

Per evitare di leggere due volte di seguito lo stesso valore, si consiglia dopo la lettura, di reinizializzare questa memoria, il comando POKE permette questa manipolazione: POKE 764,255 pone 255 (codice nullo) nella memoria numero 764.

SPOSTAMENTO DI OGGETTI DAL BASSO IN ALTO SULLO SCHERMO: SCROLL

È difficile, se si lavora in BASIC, generare lo spostamento simultaneo di vari oggetti sullo schermo (se non si ottiene velocemente una animazione accidentata, a causa di un tempo di spostamento proibitivo).

L'ATARI realizza automaticamente uno "**Scrolling**" (cioè uno spostamento dell'insieme dell'immagine) verso l'alto, quando si scrive sull'ultima riga in basso dello schermo.

Questa proprietà è stata messa a profitto in vari giochi: ciò permette, infatti, di spostare di una riga verso l'alto i vari oggetti (testi e grafici) presenti sullo schermo in quel momento. La linea in alto è evidentemente persa.

Se un oggetto deve restare immobile sullo schermo, sarà necessario cancellarlo dalla sua nuova posizione e sostituirlo alla sua posizione iniziale.

Lancio di dadi

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma proposto non realizza un gioco vero e proprio, ma potrà essere utilizzato come sottoprogramma per numerosi giochi.
Offre semplicemente la possibilità di lanciare da 1 a 9 dadi, con la semplice pressione di un tasto. I dadi sono visualizzati in seguito con una grafica molto semplificata.

IL PROGRAMMA: La linea 60 prende il carattere battuto sulla tastiera e lo trasforma in numero (se non è possibile, il programma rifiuta l'ingresso).
Una variabile D dimensionata riceve il valore indicato da ogni dado.
Un ciclo J (110-130) è utilizzato per disegnare un fondo, e un altro ciclo I (100-160) serve al lancio (lancio alla linea 140 e stampa alla linea 150).
La linea 230 offre la possibilità di rifare un lancio; non è indispensabile.

PER GIOCARE: Si lancia il programma con l'ordine RUN e si ottiene il messaggio NUMERO DI DADI DA LANCIARE? al quale si risponde battendo una cifra.
Il risultato è visualizzato automaticamente; basta battere un tasto qualunque per ricominciare. Per fermarsi, è necessario battere BREAK.

ESTENSIONI POSSIBILI: Si potrà, naturalmente, utilizzare questo programma per la realizzazione di ogni gioco utilizzante dei dadi (421, ecc.)

```
10 REM LANCIO DI DADI
15 DIM D(9)
20 OPEN #1,4,0,"K:"
30 SETCOLOR 1,13,2:SETCOLOR 2,5,14
40 PRINT "}"
50 POSITION 0,0:PRINT "Numero di dadi da lanciare :";
60 GET #1,A
70 IF A<49 OR A>57 THEN 40
80 N=A-48
90 PRINT "}"
100 FOR I=1 TO N
110 FOR J=1 TO 3
120 POSITION 4*I-4,8+J:PRINT CHR$(160);CHR$(160);CHR$(160)
130 NEXT J
140 D(I)=INT(RND(1)*6+1)
150 POSITION 4*I-3,10:PRINT D(I)
160 NEXT I
170 RESTORE :FOR I=1 TO 3
180 READ V:SOUND 0,V,10,4
190 FOR T=1 TO 10:NEXT T
200 SOUND 0,0,0,0
210 FOR T=1 TO 5:NEXT T
220 NEXT I
230 GOTO 50
1000 DATA 126,108,96
```

Tombola

Il gioco qui presentato differisce alquanto da quello praticato in Italia.

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: fortuna (vari giocatori).
Difficoltà del gioco: bassa.

IL GIOCO: Si tratta di realizzare una partita completa di tombola, cioè di ottenere 6 numeri, compresi tra 1 e 49, più un numero complementare, tutti differenti, che potrebbero essere, per esempio, utilizzati per riempire la vostra prossima cartella di tombola.

IL PROGRAMMA: Si utilizzerà un array N(7), dimensionato alla linea 40, per registrare i risultati.
La pesca dei numeri è effettuata in un ciclo I (60-120), un secondo ciclo J (90-110) è utilizzato per controllare che non ci siano due pescate identiche.
La pescata (o la ripescata) è effettuata alla linea 70.
Il ciclo I (160-190) visualizza il risultato, e la linea 200 visualizza il numero complementare.
Le linee da 220 a 230 si occupano del caso in cui si vorrebbe rifare una partita.

PER GIOCARE: Basta battere RUN per lanciare il programma e ottenere la visualizzazione di un primo numero.
La domanda UN'ALTRA ESTRAZIONE (S/N) appare nel basso dello schermo; si risponderà S se si desidera continuare.

ESTENSIONI POSSIBILI: Si può immaginare un programma che potrebbe tenere conto sia dei propri risultati, sia dei risultati inseriti da un giocatore, per migliorare la pescata, tenendo conto delle probabilità di avere un dato numero.
Si può anche immaginare un programma che realizzi dei cartoncini per la tombola come quella che è ancora correntemente praticata nel Sud della Francia.

```
10 REM TOMBOLA
20 OPEN #1,4,0,"K:"
40 DIM N(7)
50 GRAPHICS 2
60 FOR I=1 TO 7
70 N(I)=INT(RND(1)*49+1)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
140 POSITION 3,0:PRINT #6;"    TOMBOLA    "
150 POSITION 3,1:PRINT #6;"---------------"
160 FOR I=1 TO 6
170 POSITION 3*I-1,5
180 PRINT #6;N(I)
190 NEXT I
200 POSITION 0,8:PRINT #6;"complementare :";N(7);
210 RESTORE :FOR I=1 TO 3:READ V:SOUND 0,V,10,4:FOR T=1 TO 10:NEXT T:SOUND 0,0,0
,0:FOR T=1 TO 5:NEXT T:NEXT I
220 PRINT :PRINT " Un'altra estrazione (S/N) ";
230 GET #1,A:IF A=83 THEN 50
240 GRAPHICS 0
1000 DATA 126,108,96
```

La bomba

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Dovete trovare, con l'aiuto di un rivelatore speciale, un pacco radioattivo deposto da un viaggiatore male intenzionato, in una camera d'albergo. Un solo problema, di misure: l'albergo possiede un milione di camere (100*100*100).
Il vostro rilevatore visualizza, in permanenza, un segnale che vi informa sulla prossimità della bomba e il tempo che vi resta, decontabilizzato di 10 in 10.
Potrete scoprire la bomba prima dell'esplosione?

IL PROGRAMMA: Una variabile numerica R, dimensionata A 3 contiene le tre coordinate che definiscono la posizione del giocatore.
La posizione della camera contenente il pacco è definita da un' estrazione casuale di 3 variabili A, B, C, alla linea 50.
Il ciclo I (70-110) disegna il detector.
Il segnale di prossimità è calcolato alle linee 130 e la linea 140 permette la visualizzazione delle coordinate della camera da scoprire, nel caso in cui si sia perso.
La linea 150 permette la visualizzazione sia del segnale che del tempo rimanente.
La linea 160 realizza il conto alla rovescia del tempo.
La linea 170 controlla se si è riusciti a trovare o no il pacco.
Il ciclo I (210-250) permette di inserire le nuove coordinate, utilizzando il sottoprogramma posizionato alle linee da 300 a 340.
Si noteranno le referenze ai numeri di linea del programma con l'aiuto di marchi (MUSIQUE SIDODO).

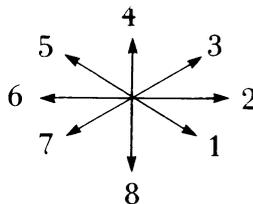
PER GIOCARE: Dopo il lancio del programma con RUN appaiono il segnale ed il tempo (all'inizio 200).
Il giocatore è allora alla posizione x=0, y=0, z=0 ; si dovranno introdurre successivamente le tre coordinate x,y,z della camera da esplorare.
Si farà, per esempio, 13 poi RETURN, 45 poi RETURN, ecc.
Il nuovo segnale è visualizzato, ed il tempo decrementato di 10.
Si continuerà così sia fino al risultato positivo (arresto del gioco e fine del programma), o fino all'apparizione di 3 numeri (x, y, z) senza uscire dal programma, il che significa che abbiamo perso.

ESTENSIONI POSSIBILI: Sarà senz'altro indispensabile migliorare la grafica, aggiungere un'esplosione (sonora e visiva) per simbolizzare l'esplosione della bomba, visualizzare simbolicamente la posizione nell'albergo, ecc.

```
10 REM LA BOMBA
15 SETCOLOR 2,3,2:POKE 752,1
20 DIM R$(10),S$(10):OPEN #1,4,0,"K:"
25 DIM R(3):FOR I=1 TO 3:R(I)=0:NEXT I
30 MUSIQUE=500
35 SIDODO=1000:DOREMI=2000:UTSOMI=3000:BEEP=4000
40 L=100:T=200
50 A=1+INT(RND(1)*(L-1)):B=1+INT(RND(1)*(L-1)):C=1+INT(RND(1)*(L-1))
60 PRINT "}"
70 FOR I=0 TO 17
80 POSITION I+10,4:PRINT CHR$(160)
90 POSITION I+10,5:PRINT CHR$(160)
100 POSITION I+10,6:PRINT CHR$(160)
110 NEXT I
120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*L))
130 S$=STR$(S):S=VAL(S$)
140 IF T=0 THEN PRINT "PERSO!. La soluzione :";A;"  ";B;"  ";C;".":RESTORE SIDOD
O:GOSUB MUSIQUE
150 POSITION 12,5:PRINT S:POSITION 22,5:PRINT T
160 RESTORE DOREMI:GOSUB MUSIQUE:T=T-INT(SQR(L))
170 IF S=0 THEN POSITION 2,20:PRINT "BRAVO!":RESTORE UTSOMI:GOSUB MUSIQUE:END
180 POSITION 0,10:PRINT "  - X :"
190 POSITION 0,12:PRINT "  - Y :"
200 POSITION 0,14:PRINT "  - Z :"
210 FOR I=1 TO 3
220 R$=""
230 GOSUB 300
240 R(I)=VAL(R$):IF R(I)>=100 THEN 220
250 NEXT I
260 GOTO 70
300 POSITION 7,2*I+8:PRINT "?";R$;:PRINT "     ";:RESTORE BEEP:GOSUB MUSIQUE
310 GET #1,CLA
320 IF CLA=155 THEN RETURN
330 R$(LEN(R$)+1)=CHR$(CLA)
340 GOTO 300
500 FOR M=1 TO 3
510 READ V:SOUND 0,V,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:FOR TE=1 TO 5:NEX
T TE
520 NEXT M
530 RETURN
1000 DATA 64,63,63
2000 DATA 63,53,47
3000 DATA 63,40,47
4000 DATA 40,40,40
```

Lavagna magica

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: disegno.
Difficoltà del gioco: bassa.

IL GIOCO: Come indica il titolo, si tratta, semplicemente, di utilizzare lo schermo come una lavagna, manipolando un punto sullo schermo seguendo 8 direzioni, con l'aiuto dei tasti da 1 a 8, secondo la convenzione:

Si dispone anche della possibilità di spostare il punto sullo schermo senza lasciare la traccia battendo 0 (il che permette la cancellazione parziale) e lasciare di nuovo la traccia ribattendo 0.
Si può scegliere il colore del punto battendo su un tasto da A a H; è possibile cancellare tutto lo schermo battendo P.

IL PROGRAMMA: La linea 30 contiene la scelta delle possibilità di colore.
La linea 50 stampa il punto .
Le linee da 60 a 90 controllano gli input.
La linea 130 chiama uno dei quattro sottoprogrammi che effettuano lo spostamento.
La linea 140 verifica che non si esca dai limiti dello schermo.
La linea 160 effettua la cancellazione se richiesta.
Il sottoprogramma 400-410 sceglie il colore; 300-310 realizza o no la cancellazione.

PER GIOCARE: Si lancia il programma con RUN; un punto appare verso la metà dello schermo. Si dovrà battere 0 per spostarlo senza visualizzarlo.
Se si batte una lettera da A a H, il colore del punto cambia. Il tasto P cancella tutto, e si ricomincia.
Si può uscire in ogni momento, battendo BREAK.

ESTENSIONI POSSIBILI: Si potrà, naturalmente, pensare di scrivere un programma identico utilizzando delle tracce di segmenti, prevedere un sottoprogramma che permetta il salvataggio del disegno realizzato per un'ulteriore utilizzazione, ecc.
Utilizzare il joystick per spostare la penna simulata.

```
10 REM LAVAGNA MAGICA
15 OPEN #1,4,0,"K:"
20 DIM C$(8)
25 GRAPHICS 11
30 C$="01234567"
40 X=40:Y=95:C=4
50 COLOR C:PLOT X,Y
60 GET #1,CLA
70 IF CLA>=ASC("A") AND CLA<=ASC("H") THEN GOSUB 400
80 IF CLA=ASC("0") THEN GOSUB 300
90 IF CLA=ASC("P") THEN GOTO 25
100 F=2*(CLA<53)-1
110 A=X:B=Y
120 U=CLA-48+2*(F-1)
130 ON U GOSUB 1000,2000,3000,4000
140 D=(X>=0 AND X<80 AND Y>=0 AND Y<192)
150 IF D=0 THEN GOSUB 5000
160 IF S<>0 THEN COLOR 0:PLOT A,B
170 GOTO 50
300 S=ABS(S-1)
310 RETURN
400 C=2*VAL(C$(CLA-64,CLA-64))
410 RETURN
1000 Y=Y+F
2000 X=X+F
2010 RETURN
3000 GOSUB 2000
4000 Y=Y-F
4010 RETURN
5000 X=A:Y=B
5010 RETURN
```

Pubblicità

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: visualizzazione pubblicitaria.
Difficoltà del gioco: nessuna.

IL GIOCO: Un testo qualunque dovrà essere introdotto all'inizio del programma. Potrà essere relativamente lungo (4 o 5 linee); si sceglierà in seguito il senso e la velocità di scorrimento, al seguito di ciò il testo apparirà (una sola linea per volta) e comincerà a scorrere, alla velocità scelta, e nel senso scelto.

IL PROGRAMMA: Il testo sarà introdotto lettera per lettera alle linee 60-90, con almeno 18 caratteri.
La visualizzazione sarà in grande formato grazie alla linea 140.
La visualizzazione sarà spezzata a 18 caratteri o più grazie alla linea 170.
Lo scorrimento verso la sinistra è preparato alla linea 180 e lo scorrimento verso destra alla linea 190.
Il testo sarà visualizzato alla linea 170 e la durata della visualizzazione (velocità di scorrimento) è scelta dalla linea 200.
La linea 220 permetterà di cambiare, se ce ne sarà bisogno, le condizioni dello scorrimento.
La linea 140 determina i diversi colori possibili.

PER GIOCARE: Si lancerà il programma con RUN e, all'apparizione della domanda INSERIRE IL TESTO, si introdurrà il testo da fare scorrere, seguito da RETURN. Si ottiene in seguito il messaggio SCORRIMENTO A DESTRA (D) O A SINISTRA (S) al quale si risponde battendo sia D, sia S.
Poi, alla domanda VELOCITÀ, si risponderà introducendo un piccolo numero per una velocità rapida, e un grande numero per una velocità lenta (per esempio, da 0 a 40 o più).
Il testo comincerà a sfilare dopo aver battuto RETURN.
Se si desiderano cambiare le condizioni dello scorrimento, basta battere il tasto 0 per ritrovare le domande SENSO e VELOCITÀ. Si batterà BREAK per interrompere il programma.

ESTENSIONI POSSIBILI: Si potrà realizzare una visualizzazione ondulante, fare risalire le linee sullo schermo, ecc.

```
10 REM PUBBLICITA'
15 DIM B$(300),C$(300),S$(1)
20 PRINT "}"
25 OPEN #1,4,0,"K:"
30 B$=""
40 PRINT "Inserite il testo :"
50 PRINT
60 GET #1,CLA
70 IF CLA=155 THEN 100
80 B$(LEN(B$)+1)=CHR$(CLA)
90 PRINT CHR$(CLA);:GOTO 60
100 PRINT :PRINT :PRINT "Scorrimento (D)estra o (S)inistra :";:INPUT S$
110 PRINT
120 PRINT "Velocita' di scorrimento (da 0 a 50) :";:INPUT V
130 V=51-V
140 GRAPHICS 2+16:FOR I=0 TO 4:SETCOLOR I,INT(RND(1)*16),2*I+1:NEXT I
150 T=LEN(B$)
160 POSITION 0,5
170 PRINT #6;" ";B$(1,18);" "
180 IF S$="S" THEN B$(T+1)=B$(1,1):B$=B$(2,T+1)
190 IF S$="D" THEN C$=B$:B$(1,1)=C$(T,T):B$(2,T)=C$(1,T-1)
200 FOR J=0 TO 10*V
210 P=PEEK(764):POKE 764,255
220 IF P=50 THEN 100
230 NEXT J
240 GOTO 160
```

Cappelli

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di riflessi, per due giocatori.
Difficoltà del gioco: bassa.

IL GIOCO: L'idea è molto semplice: due personaggi si spostano l'uno verso l'altro sullo schermo e si incrociano. Quando si incrociano, è possibile che scambino i loro cappelli. Si tratta di essere il primo a dire se c'è stato o no lo scambio di cappelli durante l'incrocio.

IL PROGRAMMA: La linea 40 contiene il disegno dei due cappelli (in due variabili alfanumeriche B$ e C$).
La linea 50 contiene, in un'altra variabile A$, i simboli che permetteranno di disegnare i personaggi.
La linea 70 permette, se è il caso, lo scambio di cappelli.
La variabile T della linea 80 serve ad individuare se c'è stato o no lo scambio.
Il disegno dei due personaggi è realizzato da un ciclo I (120-150), e il loro spostamento è realizzato dalla linea 160.
Il test, che permette o no lo scambio dei cappelli, è effettuato alla linea 180.
Le linee da 200 a 260 determinano se il primo giocatore che ha battuto su un tasto ha perso o vinto.

PER GIOCARE: Lanciare il programma con RUN.
I personaggi cominciano immediatamente a spostarsi l'uno verso l'altro.
Il giocatore di sinistra dispone dei tasti 1 e 2 e quello di destra, dei tasti 9 e 0.
Se c'è stato scambio di cappello, i tasti da premere saranno 1 (a sinistra) e 0 (a destra).
Se non c'è stato scambio, bisognerà per vincere essere i primi a premere 2 (a sinistra) e 9 (a destra).
Apparirà allora il messaggio BRAVO SINISTRA o BRAVO DESTRA, o, ancora, PERSO SINISTRA o PERSO DESTRA.

ESTENSIONI POSSIBILI: Sarà possibile prevedere una visualizzazione a colori, con una animazione dei personaggi.
Si potrà anche complicare il gioco a piacere con vari oggetti scambiati in maniera casuale (bisognerà allora trovarne la lista completa).

```
10 REM CAPPELLI
15 MUSIQUE=500:DOMISO=1000:DOREMI=2000:DODODO=3000
20 DIM A$(24),B$(3),C$(3),D$(3)
25 SETCOLOR 1,15,8:SETCOLOR 2,11,2
30 POKE 752,1
40 B$="ipo":C$="i;o":REM battere i caratteri di questa riga premendo simultaneam
ente <CONTROL>
50 A$=" t qse |  | axel || lc z":REM vedere 40 escluso | che si ottiene direttam
ente alla tastiera
60 X=8:Y=30:T=1
70 D$=C$:C$=B$:B$=D$
80 T=ABS(T-1)
90 PRINT "}"
100 POSITION X,5:PRINT C$
110 POSITION Y,5:PRINT B$
120 FOR I=1 TO 8
130 POSITION X,I+5:PRINT A$(3*I-2,3*I)
140 POSITION Y,I+5:PRINT A$(3*I-2,3*I)
150 NEXT I
160 X=X+1:Y=Y-1
165 IF Y=8 THEN RUN
170 FOR I=1 TO 200:NEXT I
180 IF X=Y AND RND(1)<0.5 THEN 70
190 P=PEEK(764):POKE 764,255:IF P=255 THEN 90
195 IF P<>30 AND P<>31 AND P<>48 AND P<>50 THEN 90
200 IF X<=Y THEN 240
220 IF (P=31 AND T=1) OR (P=30 AND T=0) THEN PRINT "BRAVO! Sinistra.":RESTORE DO
MISO:GOSUB MUSIQUE:END
230 IF (P=50 AND T=1) OR (P=48 AND T=0) THEN PRINT "BRAVO! Destra.":RESTORE DORE
MI:GOSUB MUSIQUE:END
240 IF P=30 OR P=31 THEN PRINT "PERSO Sinistra."
250 IF P=48 OR P=50 THEN PRINT "PERSO Destra."
260 RESTORE DODODO:GOSUB MUSIQUE:END
500 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:FO
R TE=1 TO 5:NEXT TE:NEXT I
510 RETURN
1000 DATA 126,96,81
2000 DATA 126,108,96
3000 DATA 126,126,126
```

Combattimento aereo

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessi (1 giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Al centro dello schermo, si trova il mirino del vostro aereo, e l'aereo nemico si sposta davanti a voi, in una maniera che potrà sembrarvi all'inizio incoerente.
Potete spostarvi in tutte le direzioni, grazie alle frecce di movimento, e dovrete cercare di portare l'aereo nemico al centro del mirino. Sparate battendo 0 e vedrete i vostri proiettili diretti verso l'aereo che, però, continua a spostarsi anche durante il vostro sparo e non è per niente facile da colpire.
La partita continuerà a svolgersi fino a che l'avrete colpito (messaggio VINTO).

IL PROGRAMMA: La linea 70 disegna l'aereo nemico il cui spostamento casuale sarà realizzato dalle linee 90 e 100.
Le linee 120-130 vi permettono il pilotaggio del vostro aereo.
La linea 170 disegna il mirino del vostro aereo con l'aiuto di caratteri grafici.
Il sottoprogramma, che gestisce lo sparo, va dalla linea 200 alla linea 250, ed il test di fine partita è effettuato alla linea 240.
La linea 180 controlla se avete sparato o no.

PER GIOCARE: Appena il programma è stato lanciato con l'ordine RUN, dovete manovrare per cercare di inquadrare l'apparecchio nemico nel vostro mirino.
Per questo, disponete delle frecce di spostamento (tasti ↑, ↓, →, ←) e potete sparare in ogni istante premendo il tasto 0.
L'apparecchio nemico può continuare a spostarsi in maniera casuale, anche quando avete sparato.
La partita terminerà con la visualizzazione di VINTO, se colpite l'aereo nemico.
Si può anche fermare in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Sarà abbastanza facile utilizzare una grafica più curata, utilizzando il colore. Si potrà anche pensare di stabilizzare la posizione dell'aereo nemico durante lo sparo.
Potremmo anche introdurre vari aerei, contare i punti od il tempo, tenere conto della scorta di carburante, ecc.

```
10 REM COMBATTIMENTO AEREO
30 GRAPHICS 0:POKE 752,1
40 SETCOLOR 1,2,2:SETCOLOR 2,8,6
50 A=5:B=15
60 N=21
70 POSITION B,A:PRINT CHR$(10);CHR$(24);CHR$(8)
80 U=A:V=B
90 A=A+2*INT(RND(1)*2)-1
100 B=B+2*INT(RND(1)*2)-1
110 P=PEEK(764):POKE 764,255
120 B=B-2*(P=6)+2*(P=7)
130 A=A-2*(P=14)+2*(P=15)
140 B=B-36*(B>36)+36*(B<0)
150 A=A-22*(A>22)+22*(A<0)
160 POSITION V,U:PRINT "   "
170 POSITION 18,9:PRINT "qwe";:POSITION 18,10:PRINT "asd";:POSITION 18,11:PRINT
"zxc";:REM battere <CONTROL>
180 IF P=50 OR N<21 THEN GOSUB 200
190 GOTO 70
200 POSITION 29-N,N:PRINT CHR$(6);:POSITION N+9,N:PRINT CHR$(7);
205 SOUND 0,31+N,10,11-INT(N/2)
210 POSITION 29-N,N:PRINT " ";:POSITION N+9,N:PRINT " ";
220 N=N-1
230 IF N>9 THEN RETURN
240 IF (A=9 OR A=10 OR A=11) AND (B=18 OR B=19 OR B=20) THEN PRINT "VINTO...":EN
D
250 GOTO 60
```

Disegno di dadi

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: bassa.

IL GIOCO: Il programma proposto è soprattutto interessante per il suo lato grafico visto che permette di raffigurare, in maniera più che discreta, da 1 a 5 dadi sullo schermo, utilizzando una tecnica di disegno un po' particolare.
Ogni risposta alla domanda NUMERO DI DADI? provoca immediatamente un nuovo lancio del numero indicato di dadi.

IL PROGRAMMA: La linea 70 permette di controllare se c'è stato o no intervento su un tasto da 1 a 5.
Le linee 90 e 100 inizializzano le due variabili di disegno dei dadi.
Il ciclo I (110-200) e la linea 120 e 130 disegnano la cornice del dado.
La linea 140 realizza il lancio, ed il ciclo J (150-190) disegna i punti in relazione al lancio realizzato (chiamando le variabili A$ e B$).

PER GIOCARE: Si lancia il programma con RUN, e alla domanda NUMERO DI DADI, si risponde battendo un tasto da 1 a 5. I dadi richiesti sono allora raffigurati, e la domanda NUMERO DI DADI viene posta di nuovo.
Basterà battere BREAK per uscire dal programma.

ESTENSIONI POSSIBILI: Si potrà conservare la parte disegno utilizzata in questo programma per realizzare tutta una serie di giochi con i dadi, con una rappresentazione grafica di qualità discreta e facile da modificare se ce ne fosse bisogno.

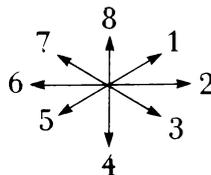
```
10 REM DISEGNO DI DADI
20 OPEN #1,4,0,"K:"
30 DIM A$(30),B$(18)
40 GRAPHICS 0:SETCOLOR 1,7,2:SETCOLOR 2,11,10
50 POSITION 0,15:PRINT "Numero di dadi :";
60 GET #1,CLA
70 IF CLA<ASC("1") OR CLA>ASC("5") THEN 40
80 PRINT "}"
90 A$="         ":A$(LEN(A$)+1)="t":A$(LEN(A$)+1)="  t          ":A$(LEN(A$)+1)="tt
  t":REM vedere 120
100 B$="121314324515525555"
110 FOR I=1 TO CLA-48
120 POSITION 8*I-8,2:PRINT "qrrrrre":REM battere simultaneamente <CONTROL>
130 POSITION 8*I-8,8:PRINT "zrrrrrc":REM voir 120
140 X=INT(RND(1)*6+1)
150 FOR J=1 TO 5
160 IF J<=3 THEN S=5*VAL(B$(3*(X-1)+J,3*(X-1)+J))-4
170 IF J<=3 THEN POSITION 8*I-7,2*J+1:PRINT A$(S,S+4)
180 POSITION 8*I-8,J+2:PRINT "|":POSITION 8*I-2,J+2:PRINT "|"
190 NEXT J
200 NEXT I
210 GOTO 50
```

Il gioco del missile

RIFERIMENTI:
Programmazione: facile.
Categoria del gioco: contro l'ATARI.
Difficoltà del gioco: media.

IL GIOCO:
L'ATARI è in allarme: siete ai comandi di un aereo incaricato di una missione di spionaggio sul suo territorio; il vostro scopo è di sorvolare la base nemica (raffigurata da un'H ingrandita), e di ritornare sani e salvi alla vostra base; ma, appena l'ATARI vi avrà avvistati, manderà al vostro inseguimento, un missile molto perfezionato, che non avrà pace fino a che non vi avrà distrutto.
Il vostro aereo può spostarsi in tutte le direzioni, grazie ai tasti da 1 a 8; riuscirete a portare a termine la vostra missione?

IL PROGRAMMA:
La linea 50 disegna la base nemica da sorvolare (sottoforma di un'H ingrandita).
La linea 60 disegna la vostra base di partenza.
Le linee 80 e 100 scelgono l'istante di partenza del missile.
La linea 120 disegna il vostro aereo.
Le linee 140 e 150 definiscono il suo movimento seguendo 8 direzioni, utilizzando i sotto-programmi alle linee 1000, 2000, 3000 e 4000.
La linea 160 assicura la partenza del missile che è disegnato dalla linea 170 ed il cui spostamento è gestito dalle linee 180 e 190.
Le linee 200 e 210 controllano la fine della partita.

PER GIOCARE:
Lanciare il programma con RUN.
Il campo di gioco appare e potete spostare il vostro aereo utilizzando i tasti da 1 a 8, con la convenzione seguente:

Dovete farlo girare attorno al territorio nemico (se voi, o il missile, toccate questo territorio, comincia una seconda partita).
Se il missile vi distrugge, ottenete un messaggio COLPITO, e la partita si ferma.
Battere BREAK per fermare la partita.

ESTENSIONI POSSIBILI: Sarà senz'altro necessario aggiungere vari controlli nel programma (l'aereo ha sorvolato la base nemica, è tornato alla propria base?...) Si possono anche prevedere vari livelli di gioco facendo, per esempio, variare la velocità del missile, prevedere l'utilizzo di trucchi per l'aereo, l'esplosione del missile dopo un certo tempo, ecc.

```
10 REM IL GIOCO DEL MISSILE
20 OPEN #1,4,0,"K:"
30 GRAPHICS 0:M=0
40 POKE 752,1
50 POSITION 17,5:PRINT CHR$(1);CHR$(4)
60 POSITION 0,20:PRINT "XXXX":POSITION 0,21:PRINT "XXXX"
70 U=15:X=19
80 S=INT(RND(1)*U+6)
90 Y=1
100 S=S-1
110 V=33
120 POSITION Y,X:PRINT "Y"
130 GET #1,CLA
140 F=2*(CLA<ASC("5"))-1
150 ON CLA-48+2*(F-1) GOSUB 1000,2000,3000,4000
160 IF S<>0 THEN 100
170 POSITION V,U:PRINT "<"
180 U=U-SGN(U-X)
190 V=V-SGN(V-Y)
200 IF U-X AND V=Y THEN PRINT "COLPITO...":END
210 IF (U=5) AND (V=17 OR V=18) THEN 30
220 IF X<=4 AND Y>=19 THEN M=1
230 IF M=1 AND X=19 AND Y<=4 THEN PRINT "Missione compiuta":END
240 GOTO 120
1000 X=X-F
2000 Y=Y+F
2010 RETURN
3000 GOSUB 2000
4000 X=X+F
4010 RETURN
```

La scimmia

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: due giocatori.
Difficoltà del gioco: elevata.

IL GIOCO: Si ispira all'idea già vecchia della scimmia che, battendo a caso su una macchina da scrivere, ha delle possibilità non nulle di finire per scrivere un testo sensato.
Qui il ruolo della scimmia è preso dall'ATARI che farà apparire dei gruppi di 3 a 6 lettere durante un tempo relativamente breve.
Se uno dei giocatori stima di poter riuscire a comporre una parola con le lettere proposte, deve battere rapidamente sul suo tasto di gioco (1 o 0).
La parola dovrà allora essere accettata o rifiutata dall'avversario che risponderà GIUSTO o SBAGLIATO.
Ogni parola giusta vale 1 punto, ogni parola sbagliata leva 1 punto.

IL PROGRAMMA: La linea 40 contiene una lista V$ delle vocali correnti e una lista C$ delle consonanti correnti.
Sarà possibile modificare queste due liste per aumentare, per esempio,la frequenza di apparizione di certe lettere (potremo così mettere due E in V$, o due R in C$, ecc.).
La linea 50 definisce la lunghezza della parola (in principio , da 3 a 6 lettere, ma si può ugualmente modificare questa lunghezza).
Il ciclo I (80-130) permette la creazione di una "parola" prendendo una lettera, sia nell'alfabeto completo, sia nella lista C$ (linea 100), sia nella lista V$ (linea 110).
La linea 150 temporizza l'apparizione del risultato.
La linea 170 gestisce la fine della partita, con visualizzazione del punteggio.
La linea 180 guarda se uno dei due giocatori ha reagito.
Infine, le linee 200 e 210 contabilizzano i punti di ogni giocatore.

PER GIOCARE: Si lancia il programma con RUN.
La prima serie di lettere è visualizzata sullo schermo. Il giocatore di sinistra dovrà battere 1 se vede che può realizzare una parola, quello di destra dovrà battere 0.
Se non è stato battuto niente dopo un po' di tempo un'altra serie di lettere sarà visualizzata.

Se uno dei giocatori ha battuto 0 o 1, appare allora la domanda (G)IUSTO o (S)BAGLIATO . Si risponde G se la parola è corretta, S se è sbagliata, seguito da RETURN.
I punti sono calcolati dall'ATARI che visualizza la serie di lettere seguente.
Per uscire dal programma si dovrà battere P al posto di 0 o 1; viene visualizzato il punteggio finale .

ESTENSIONI POSSIBILI: Si potrà modificare la lunghezza delle parole, la frequenza di apparizione delle lettere, il conteggio dei punti, la regola del gioco, ecc.

```
10 REM LA SCIMMIA
15 OPEN #1,4,0,"K:"
20 DIM C$(12),E$(1),R$(1),V$(5),M$(6)
30 MUSIQUE=500:DOREMI=1000:DOMISO=2000
40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPRSTV"
50 X=INT(RND(1)*4+3)
60 M$=""
70 GRAPHICS 2:FOR I=0 TO 4:SETCOLOR I,INT(RND(1)*16),2*(I+1):NEXT I
80 FOR I=1 TO X
90 E$=CHR$(INT(RND(1)*26+65))
100 IF RND(1)<0.4 THEN V=INT(LEN(C$)*RND(1)+1):E$=C$(V,V)
110 IF RND(1)<0.5 THEN V=INT(LEN(V$)*RND(1)+1):E$=V$(V,V)
120 M$(LEN(M$)+1)=E$
130 NEXT I
140 POSITION 0,5:PRINT #6;M$:RESTORE DOREMI:GOSUB MUSIQUE
150 FOR J=1 TO 500:NEXT J
160 CLA=PEEK(764):POKE 764,255
170 IF CLA=10 THEN PRINT "  Punti Sinistra :";B;" - Punti Destra :";A:RESTORE DO
MISO:GOSUB MUSIQUE:END
180 IF CLA<>50 AND CLA<>31 THEN 50
190 PRINT "}":PRINT "     (G)iusto o (S)basliato :";:INPUT R$
200 A=A+(R$="G" AND CLA=50)-(R$="S" AND CLA=50)
210 B=B+(R$="G" AND CLA=31)-(R$="S" AND CLA=31)
220 GOTO 50
500 FOR I=1 TO 3
510 READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:FOR TE=1 TO 5:N
EXT TE
520 NEXT I
530 RETURN
1000 DATA 126,108,96
2000 DATA 126,96,81
```

Il gioco del 15

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO: È un gioco per due giocatori, abbastanza vecchio, che era un tempo praticato nelle fiere. Si dispone di un tappeto di gioco di 9 caselle, e ognuno scommette a turno un gettone del suo colore, su una casella disponibile.
Il primo dei due giocatori che, con tre gettoni esattamente, riesce a realizzare un totale di 15 ha vinto.

IL PROGRAMMA: Le linee I 50-80 disegnano il tavolo di gioco, e le linee 90-100 si occupano della gestione del turno dei giocatori (numero 1 o 2 per ogni giocatore, e gettone O o X corrispondente).
La linea 110 attende il colpo giocato.
Il ciclo I (120-140) verifica la validità del colpo.
La linea 150 testa la fine della partita, in caso di partita nulla, e la linea 160 rifiuta i colpi non regolamentari (con ritorno alla situazione iniziale, linea 90).
Le linee 170 e 180 si occupano della visualizzazione del colpo, e la linea 190 lo registra in una variabile R$. La parte più interessante del programma, che potrebbe servire per definire una strategia per la macchina, va dalla linea 210 alla linea 300: si utilizzano 3 cicli chiusi per verificare se uno dei giocatori ha ottenuto, con 3 gettoni, un totale di 15. Il cambio del giocatore si effettua alla linea 310.

PER GIOCARE: Lanciare il programma battendo RUN.
Il campo di gioco appare, poi la domanda GIOCATORE 1: X, e il giocatore, che ha scelto le X, dovrà dare il numero della casella dove desidera giocare; se è libera, il suo colpo vi sarà registrato.
Si ottiene, in seguito, il messaggio GIOCATORE 2: O ed il secondo giocatore procede nella stessa maniera.
Se uno dei giocatori vince (totale di 15 in tre colpi), ottiene il messaggio BRAVO e la partita si ferma.
In caso di partita pari, la macchina visualizza È FINITA.
Si farà un'altra partita battendo RUN.

ESTENSIONI POSSIBILI: La più interessante sarà senz'altro la definizione di una strategia per la macchina (cercando di evitare che la macchina giochi sempre il 5 quando deve cominciare).

Si può, naturalmente, migliorare la grafica e introdurre effetti sonori per i diversi messaggi.
Si potranno modificare la lunghezza delle parole, la frequenza di apparizione delle lettere, il conteggio dei punti, la regola del gioco, ecc.

```
10 REM IL GIOCO DEL 15
15 DIM O$(9),X$(9),R$(9),T$(9)
20 GRAPHICS 0:SETCOLOR 1,3,2:SETCOLOR 2,8,10:POKE 752,1
25 OPEN #1,4,0,"K:"
30 F=0:R$="123456789"
35 MUSIQUE=500:DODODO=1000:DOMISO=2000
40 O$="":X$=""
50 FOR I=0 TO 24 STEP 3
60 POSITION I+5,1:PRINT CHR$(17);CHR$(18);CHR$(5):POSITION I+5,2:PRINT "| |"
65 POSITION I+5,3:PRINT CHR$(26);CHR$(18);CHR$(3)
70 POSITION I+6,4:PRINT (I+3)/3
80 NEXT I
90 POSITION 0,10:PRINT "Giocatore ";F+1;
100 IF F=1 THEN PRINT " : O"
105 IF F<>1 THEN PRINT " : X"
110 GET #1,CLA:IF CLA<ASC("1") OR CLA>ASC("9") THEN 110
120 FOR I=1 TO LEN(R$)
130 IF CHR$(CLA)=R$(I,I) THEN GOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THEN POSITION 0,20:PRINT "E' finita....":RESTORE DODODO:GOSUB
MUSIQUE:END
160 GOTO 90
170 IF F=1 THEN O$(LEN(O$)+1)=CHR$(CLA):POSITION 3*I+3,2:PRINT "O"
180 IF F=0 THEN X$(LEN(X$)+1)=CHR$(CLA):POSITION 3*I+3,2:PRINT "X"
190 R$(I,I)="0"
200 IF F=1 THEN T$=O$
205 IF F<>1 THEN T$=X$
210 IF LEN(T$)<3 THEN 310
220 FOR I=1 TO 3
230 IF I=LEN(T$) THEN 310
240 FOR J=I+1 TO LEN(T$)
250 IF J=LEN(T$) THEN 310
260 FOR K=J+1 TO LEN(T$)
270 IF VAL(T$(I,I))+VAL(T$(J,J))+VAL(T$(K,K))=15 THEN 330
280 NEXT K
290 NEXT J
300 NEXT I
310 F=ABS(F-1)
320 RETURN
330 POSITION 0,10
340 PRINT "Bravo giocatore ";:IF F=1 THEN PRINT "O.":GOTO 360
350 PRINT "X."
360 RESTORE DOMISO:GOSUB MUSIQUE
370 END
500 FOR I=1 TO 3
510 READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:FOR TE=1 TO 5:N
EXT TE:NEXT I
520 RETURN
1000 DATA 63,63,63
2000 DATA 126,96,81
```

Destrezza

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessi (1 giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Delle cifre, tra 0 e 9, appariranno a caso, in diverse locazioni dello schermo. Restano visualizzate per un tempo molto corto. Il giocatore deve riuscire a battere sul tasto numerico corrispondente, nel momento in cui appare la cifra. La cifra battuta dal giocatore sarà allora visualizzata in grande formato sulla sinistra dello schermo; ed ogni risposta esatta dà 10 punti. La durata della partita è casuale ed il punteggio apparirà alla fine della stessa.

IL PROGRAMMA: Le linee 50 e 60 contengono, in due variabili alfanumeriche C$ e B$, i riferimenti che saranno utilizzati per disegnare le cifre in grande formato.
La tabella di conversione dei caratteri è contenuta nella stringa T$.
La posizione della cifra presa dall'ATARI è scelta alle linee 70 e 80, e la cifra è presa alla linea 90.
La linea 100 realizza la stampa, e la linea 110 determina la durata della visualizzazione. La linea 130 definisce la durata del gioco, e la fine della partita è trattata alla linea 150.
La linea 192 verifica che il tasto battuto sia una cifra da comparare con i codici tastiera delle cifre contenute in T$.
Il ciclo I (200-230) realizza il disegno grande formato della cifra battuta dal giocatore.
Il controllo dell'esattezza del risultato è effettuato dalla linea 240.

PER GIOCARE: Il programma sarà lanciato con RUN, poi una prima cifra appare, dopo una breve pausa, verso il centro dello schermo. Il giocatore dovrà reagire abbastanza in fretta, intanto che rimane la visualizzazione, e ribatterla.
La cifra che ha battuto si disegnerà allora in alto a sinistra dello schermo, in grande formato.
Se il risultato è esatto, il giocatore segnerà 10 punti.
Il tempo di svolgimento della partita dipenderà dalle cifre che saranno visualizzate.
Alla fine della partita apparirà il punteggio.
Sarà possibile fermarsi in ogni istante battendo BREAK.

ESTENSIONI POSSIBILI: Si può immaginare di modificare il gioco per fare partecipare vari giocatori, modificando eventualmente la visualizzazione e codificando i tasti della tastiera messi a disposizione di ogni giocatore.
Si potranno anche levare dei punti ad ogni risultato sbagliato, modificare le regole di calcolo del punteggio e la durata del gioco, ecc.

```
10 REM DESTREZZA
20 DIM B$(50),C$(12),T$(20)
25 SETCOLOR 1,8,6
30 POKE 752,1
40 T=100:S=0:PRINT "}"
50 C$="****     ** *":REM battere uno spazio con colore inverso al posto di ogni
*
60 B$="14441333313121131314413312131121411333314141114131"
65 T$="50313026242927515348"
70 X=INT(RND(1)*15)
80 Y=INT(RND(1)*15+15)
90 N=INT(RND(1)*10)
100 POSITION Y,X:PRINT N
110 FOR I=0 TO RND(1)*350+200:NEXT I
120 P=PEEK(764):POKE 764,255:IF P<>255 THEN 190
130 T=T-N
140 POSITION Y,X:PRINT "
150 IF T>=0 THEN 70
160 PRINT "}"
170 PRINT :PRINT "il tuo punteggio e' di ";S;" su 100."
180 FOR T=1 TO 255:SOUND 0,T,10,INT(T/16):NEXT T:END
190 FOR I=0 TO 9
192 IF T$(2*I+1,2*I+2)<>STR$(P) THEN NEXT I
194 U=I
200 FOR I=1 TO 5
210 A=3*VAL(B$(5*U+I,5*U+I))-2
220 POSITION 2,2*I:PRINT C$(A,A+2):POSITION 2,2*I+1:PRINT C$(A,A+2)
230 NEXT I
240 IF U=N THEN S=S+10
270 GOTO 130
```

Sottomarino

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: riflessi (1 solo giocatore).
Difficoltà del gioco: bassa.

IL GIOCO: Manovrate in superficie un cacciasommergibili e dei sottomarini passano regolarmente sotto la vostra unità, a differenti profondità.
Potete, in ogni momento, lanciare le bombe di profondità e potete anche cercare di colpire più volte lo stesso sommergibile.
Il vostro punteggio sarà visualizzato in permanenza sulla sinistra dello schermo (rapporto di numero di colpi centrati su numero di sottomarini apparsi).

IL PROGRAMMA: La linea 70 permette all'ATARI di scegliere la profondità del sottomarino.
La linea 90 disegna la vostra nave.
Le linee 100 e 110 visualizzano il punteggio e disegnano la superficie dell'acqua.
Là linea 120 disegna il sottomarino.
Il lancio è effettuato da un programma specifico posizionato alle linee 200-240.
Lo spostamento (molto rapido) della vostra nave è realizzato dalle linee 170 e 180.
Il test, che permette di sapere se il bersaglio è stato centrato o no, è effettuato alla linea 140.

PER GIOCARE: Battere RUN per cominciare la partita.
La nave sarà spostata grazie alle frecce destra e sinistra.
Il tasto O autorizza il tiro.
Per terminare la partita, converrà battere BREAK oppure aspettare 99 sottomarini.

ESTENSIONI POSSIBILI: Sarà abbastanza elementare migliorare la qualità grafica del gioco, utilizzare il colore e i suoni, fare apparire vari sottomarini che si spostano a differenti velocità, eventualmente anche permettere ai sottomarini di lanciare dei siluri...

```
10 REM SOTTOMARINO
20 POKE 752,1
30 MUSIQUE=500:DODODO=1000:DOREMI=2000
40 P=0:C=15:S=5
50 PRINT "}":SETCOLOR 1,7,2:SETCOLOR 2,8,6
60 POSITION 0,S:PRINT "       "
70 T=27:S=INT(RND(1)*8+4):P=P+0.01
80 X=1
90 POSITION C,2:PRINT "    ";CHR$(138);CHR$(160);CHR$(153);"    "
100 POSITION 0,3:FOR I=0 TO 19:PRINT CHR$(6);CHR$(7);:NEXT I
110 POSITION 0,18:PRINT "Puntessio :";P
120 POSITION T,S:PRINT CHR$(18);CHR$(160);CHR$(18);" "
130 IF X=1 THEN Y=C+2*X
140 IF S=X AND (T+1=Y OR T+2=Y) THEN P=P+1:POSITION T,S:PRINT "***":RESTORE DODO
DO:GOSUB MUSIQUE
150 T=T-1:IF P>=99 THEN POSITION 0,18:PRINT "Finito...":RESTORE DOREMI:GOSUB MUS
IQUE:END
160 IF T<0 THEN POSITION Y,X:PRINT " ":GOTO 60
170 CLA=PEEK(764):POKE 764,255
180 C=C+3*((CLA=7 AND C<27)-(CLA=6 AND C>0))
190 IF CLA<>50 AND X=1 THEN 90
200 POSITION Y,X
210 PRINT " ":X=X+1
220 IF X>S THEN 80
230 POSITION Y,X:PRINT "O"
240 GOTO 90
500 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 5:NEXT TE:SOUND 0,0,0,0:FOR
 TE=1 TO 3:NEXT TE:NEXT I:RETURN
1000 DATA 63,63,63
2000 DATA 126,108,96
```

Zombi

RIFERIMENTI: Programmazione: abbastanza difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: bassa.

IL GIOCO: Gli ZOMBI sono degli esseri malefici che cercheranno di distruggervi, ma anche voi cercherete di utilizzarli a vostro vantaggio, per sbarazzare il vostro spazio vitale dai buchi neri che l'oscurano. Sullo schermo è disegnato un perimetro che rappresenta il vostro spazio vitale. In questo perimetro, appariranno in maniera casuale gli ZOMBI (sono delle Z invertite), i buchi neri (dei #) ed il giocatore (una O). Gli ZOMBI convergono verso il giocatore per cercare di catturarlo, ma lui si sposta più rapidamente di loro e cercherà di far loro ingoiare i buchi neri, senza farsi prendere.
La partita termina sia con la sparizione del giocatore, sia con l'eliminazione di tutti gli ZOMBI.

IL PROGRAMMA: La posizione degli ZOMBI è definita da un array Z dimensionato.
Le linee da 50 a 120 (ciclo I) disegnano il perimetro, gli ZOMBI e i buchi neri.
La linea 130 definisce la posizione iniziale del giocatore.
Un secondo ciclo I (150-240) realizza lo spostamento del giocatore.
La linea 210 legge il carattere direttamente nella memoria dello schermo per sapere la natura dell'ostacolo che si trova in Z(I,+4),Z(I).
Le linee da 290 a 340 controllano lo spostamento del giocatore.

PER GIOCARE: Dopo aver lanciato l'esecuzione con RUN, il giocatore dovrà spostarsi utilizzando le 4 frecce e cercando di non uscire dai limiti del terreno. Dovrà manovrare in modo di fare andare gli ZOMBI sui buchi neri, senza farsi lui stesso bloccare.
Per fermare la partita, basterà battere BREAK.

ESTENSIONI POSSIBILI: Sarebbe interessante ingrandire il campo di gioco, aumentare il numero degli ZOMBI o dei buchi neri, migliorare i controlli, spostare i giocatori in modo più casuale, eventualmente modificare la regola.
Si potrà anche, naturalmente, migliorare la qualità del disegno.

```
10 REM ZOMBI
15 OPEN #1,4,0,"K:"
20 DIM S$(1),Z(10)
25 FOR I=1 TO 10:Z(I)=0:NEXT I
30 MUSIQUE=500:DOREDO=1000:DOMISO=2000:DOREMI=3000
40 GRAPHICS 0:POKE 752,1
45 SETCOLOR 1,3,6:SETCOLOR 2,13,4
50 FOR I=0 TO 21
60 POSITION I+1,0:PRINT CHR$(160)
70 POSITION 0,I:PRINT CHR$(160):POSITION 23,I:PRINT CHR$(160)
80 IF RND(1)<0.5 THEN POSITION I,INT(RND(1)*20+1):PRINT "#"
90 S=INT(RND(1)*14+1)
100 Z(INT(I/3)+1)=5+S
110 POSITION 1+I,21:PRINT CHR$(160)
120 NEXT I
130 K=INT(RND(1)*18+1)
140 P=0
150 FOR I=1 TO 4
160 P=P+Z(I)
170 IF Z(I)=0 THEN 240
175 PRINT
180 POSITION Z(I+4),Z(I):PRINT " "
190 Z(I)=Z(I)-SGN(Z(I)-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
210 LOCATE Z(I+4),Z(I),EC:POSITION Z(I+4),Z(I):PRINT CHR$(EC);:IF EC=ASC("#") TH
EN Z(I)=0:GOTO 230
220 POSITION Z(I+4),Z(I):PRINT "Z"
230 IF Z(I)=S AND Z(I+4)=K THEN POSITION 0,23:PRINT "Hai perso ...";:RESTORE DOR
EDO:GOSUB MUSIQUE:GOTO 350
240 NEXT I
250 IF P=0 THEN POSITION 0,23:PRINT "BRAVO! tutti gli zombi sono morti";:RESTORE
 DOMISO:GOSUB MUSIQUE:GOTO 350
260 POSITION K,S:PRINT "O"
270 FOR I=1 TO 1200:NEXT I
280 POSITION K,S:PRINT " "
290 CLA=PEEK(764):POKE 764,255
300 K=K-2*((CLA=6)-(CLA=7))
310 IF K<=0 OR K>=23 THEN POSITION 0,23:PRINT "Peccato!...";:RESTORE DOREMI:GOSU
B MUSIQUE:GOTO 350
320 S=S-2*((CLA=14)-(CLA=15))
330 IF S<=0 OR S>=21 THEN K=25:GOTO 310
340 GOTO 140
350 GET #1,CLA:END
500 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 5:NEXT TE:SOUND 0,0,0,0:FOR
 TE=1 TO 3:NEXT TE:NEXT I:RETURN
1000 DATA 126,108,126
2000 DATA 63,47,40
3000 DATA 126,108,96
```

Diago

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: riflessi (1 solo giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Si lancia il programma con RUN e si ottiene un primo messaggio che vi informa sulla prestazione da realizzare (per esempio, 5 punti neri o 17 punti neri...), poi il tempo è visualizzato e decontabilizzato, a partire da 10 secondi. Il gioco comincia quando si arriva a 0 secondi. Dei cerchi appaiono allora in basso a sinistra dello schermo, formando una linea diagonale dovrete battere il tasto 0 per fermare la visualizzazione dei punti appena stimerete che il numero di punti richiesti all'inizio è apparso sullo schermo.
Si ottiene allora il messaggio BRAVO o PERSO secondo il risultato ottenuto.

IL PROGRAMMA: Il numero di punti neri sarà scelto a caso, alla linea 50.
Il tempo è calcolato dalle linee 60-120.
I punti sono stampati e la linea diagonale è disegnata dalla linea 140.
La pressione del tasto 0 è rivelata alla linea 170, e la linea 190 controlla la fine della partita (nessun riflesso).
I messaggi finali sono eseguiti dalle linee da 190 a 230.

PER GIOCARE: Si lancerà l'esecuzione con RUN.
Si ottiene allora la visualizzazione della frase IL BUON RIFLESSO È A ... che indica il numero di punti vincenti.
Il tempo sarà in seguito decontabilizzato sotto forma ATTENZIONE: RESTANO ... SECONDI che indica il tempo rimanente prima dell'apparizione della diagonale.
Alla visualizzazione di 0, la diagonale appare e comincia a spostarsi sullo schermo.
Se si ritiene che il numero di punti richiesti è stato raggiunto, si dovrà premere il tasto 0.
Apparirà allora BRAVO se si vince, o PERSO nel caso contrario; poi la frase ALTRA PROVA (S/N) appare, rispondere S per fare un'altra partita.

ESTENSIONI POSSIBILI: Si potrebbe pensare ad un adattamento del gioco a due o più giocatori, lasciando che la diagonale si formi completamente e testarne la coincidenza alla pressione del tasto assegnato al giocatore.

Si può anche realizzare una grafica più curata e modificare il principio dello spostamento degli oggetti sullo schermo (spostamento orizzontale, verticale o apparizione a caso...).

```
10 REM DIAGO
20 MUSIQUE=500:SOMIDO=1000:DOREMI=2000
30 OPEN #1,4,0,"K:"
40 PRINT "}"
50 X=INT(RND(1)*18+3)
60 S=10
70 PRINT "       Il buon riflesso e' a ";X
80 POSITION 4,10:PRINT "Attenzione! restano ";S;" secondi.";
90 S=S-1
100 FOR I=1 TO 5:SOUND 0,126,10,4:NEXT I:FOR I=1 TO 400:NEXT I
110 PRINT "}"
120 IF S<>0 THEN 70
130 U=0
140 POSITION U,23:PRINT "O"
150 FOR I=1 TO 30:NEXT I
160 U=U+1
170 P=PEEK(764):POKE 764,255
180 IF P=50 THEN 210
190 IF U>22 THEN PRINT "Perso!":RESTORE SOMIDO:GOSUB MUSIQUE:GOTO 230
200 GOTO 140
210 IF U<>X THEN U=23:GOTO 190
220 PRINT "BRAVO!":RESTORE DOREMI:GOSUB MUSIQUE
230 PRINT :PRINT "Un'altra prova (S/N) ";
240 GET #1,CLA:IF CLA=ASC("S") THEN 40
250 END
500 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR T=1 TO 10:NEXT T:SOUND 0,0,0,0:FOR
T=1 TO 5:NEXT T:NEXT I:RETURN
1000 DATA 81,96,126
2000 DATA 63,53,47
```

Recomat

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: soluzione del programma Nicomat.
Difficoltà del gioco: nessuna.

IL GIOCO: È una replica del programma Nicomat, proposto nelle pagine precedenti; inserite i resti successivamente delle divisioni per 3, 5 e 7 di un intero compreso tra 1 e 100, e l'ATARI calcola e visualizza questo intero. Il metodo usato può d'altronde essere generalizzato (a voi trovare come).

IL PROGRAMMA: Le linee 30-50 visualizzano qualche linea di spiegazione (gli spazi che mancano apparentemente sono voluti).
La linea 50 aspetta che battiate un resto, che sarà rifiutato se è incompatibile con la divisione col numero visualizzato.
La linea 100 immagazzina i resti in una variabile non dimensionata A (S).
La linea 190 rivisualizza i resti inseriti.
Le linee 210-230 gestiscono la fine della partita.

PER GIOCARE: Si lancia il programma con RUN. Appare un testo di presentazione poi il messaggio "RESTO PER: 3, 5, 7" appare.
Si deve rispondere e la soluzione è visualizzata appena si sono dati i tre resti.
Per un'altra partita, battere S alla domanda UN'ALTRA PROVA (S/N).
Battere N per finire.

ESTENSIONI POSSIBILI: Si può estendere il programma agli interi da 1 a 1000, utilizzando i resti per 3, 5, 7, e 11; si può aggiungere il suono ed il colore per rendere il programma più attraente.

```
10 REM RECOMAT
15 OPEN #1,4,0,"K:":GRAPHICS 2
20 DIM A(3)
25 POKE 752,1
30 PRINT "INSERITE I TRE RESTI DELLE DIVISIONI DI UN NUMERO INTERO INFERIORE A 1
00, PER 3,5 E 7."
60 I=3:S=1
70 POSITION 3,0:RESTORE :GOSUB 500
80 PRINT #6;"RESTE / PAR: ";I
85 POSITION 3,1:PRINT #6;"--------------"
87 POSITION 12,4:PRINT #6;"-----"
90 POSITION 0,5:PRINT #6;"RESTO PER:  3 5 7":REM battere la stringa da stampare
in caratteri inversi
95 GET #1,CLA:IF CLA<ASC("0") OR CLA>=ASC(STR$(I)) THEN 95
100 A(S)=CLA-48
105 POSITION 9+I,3:PRINT #6;A(INT(I/2))
110 S=S+1:I=I+2
120 IF I<8 THEN 70
130 T=A(1)*70+A(2)*21+A(3)*15
200 IF T>105 THEN T=T-105:GOTO 200
210 GRAPHICS 2+16+32:POSITION 0,8:PRINT #6;"soluzione :";T
220 FOR T=1 TO 1000:NEXT T:GRAPHICS 2+32:PRINT "   Un'altra prova (S/N) ";:GET #
1,CLA
230 IF CLA=ASC("S") THEN RUN
240 END
500 FOR IT=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:F
OR TE=1 TO 5:NEXT TE:NEXT IT:RETURN
1000 DATA 63,53,47
```

Caccia al sottomarino

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: media.

IL GIOCO: Siete a bordo di una nave, incaricata di scovare un sommergibile nascosto da qualche parte nel vostro settore; dovete spostarvi fino ad arrivare sopra al sottomarino e sparare ad una certa profondità; ma attenzione: il sommergibile è anch'esso mobile; e spegnerà i suoi motori solo quando sarete esattamente sulla verticale, per non essere scoperto.

IL PROGRAMMA: Il ciclo I (60-120) visualizza il terreno di gioco.
La linea 140 definisce la posizione iniziale del sottomarino; la linea 150 permette il suo spostamento, controllato dalla linea 160 (si resta nei limiti del campo).
La linea 180 visualizza la vostra posizione.
Le linee 162-166 controllano il vostro spostamento.
La distanza è calcolata alla linea 250, visualizzata alla linea 260 e testata alle linee 270 e 280.
Il test di fine partita è effettuato alla linea 310.

PER GIOCARE: Si lancia il programma con RUN e si inserisce la dimensione; appare la griglia di gioco. Siete all'inizio alla posizione (0,0) e vi spostate con le frecce.
Dopo ogni spostamento appare la nuova distanza del sottomarino; ad ogni colpo il sottomarino si sposta.
Se riuscite a piazzarvi sulla verticale, la distanza 0.
Sparate dando una profondità intera (tra 2 e 4, limiti inclusi). Se lo mancate, si riposiziona in un posto qualunque, e si ricomincia; se lo colpite, ottenete il messaggio BRAVO e la partita termina battendo nuovamente alla tastiera.
Si puo fermare il programma in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: È possibile prevedere un miglioramento della grafica, migliorare le possibilità di spostamento del caccia (un po' lento, in questa versione), prevedere dei percorsi più sofisticati per il sottomarino, di mettere un contatore di tempo...

```
10 REM CACCIA AL SOTTOMARINO
20 OPEN #1,4,0,"K:"
40 PRINT "Dimensione della griglia :";:INPUT H
45 POKE 752,1
50 PRINT "}"
60 FOR I=0 TO H
70 POSITION 2*I+4,0:PRINT I
80 POSITION 1,2*I+2:PRINT I
90 FOR J=0 TO H
100 POSITION 2*J+4,2*I+2:PRINT "."
110 NEXT J
120 NEXT I
130 I=1:X=0:Y=0:POSITION 4,2:PRINT CHR$(160)
140 V=INT(RND(1)*H):U=INT(RND(1)*H)
150 U=U+I:V=V+I
160 IF (U<0 OR U>H OR V<0 OR V>H) THEN I=-I:GOTO 150
162 GET #1,CLA
164 Y=Y-(CLA=43 AND Y>0)+(CLA=42 AND Y<H)
166 X=X-(CLA=45 AND X>0)+(CLA=61 AND X<H)
170 POSITION 2*Y+2,2*X+2:IF Y<>0 THEN PRINT "."
180 POSITION 2*Y+4,2*X+2:PRINT CHR$(160)
190 POSITION 2*Y+6,2*X+2:IF Y<>H THEN PRINT "."
200 POSITION 2*Y+4,2*X:IF X<>0 THEN PRINT "."
210 POSITION 2*Y+4,2*X+4:IF X<>H THEN PRINT "."
250 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
260 POSITION 29,10:PRINT "Dist. ";STR$(D);" ";
270 IF D=0 THEN 300
280 IF D<3 THEN 150
290 GOTO 162
300 POSITION 29,12:PRINT "Prof. ";:INPUT D
310 IF D=INT(RND(1)*3+2) THEN POSITION 29,15:PRINT "BRAVO!":GET #1,CLA:END
320 FOR TE=0 TO 5:SOUND 0,126,10,4*(TE/2=INT(TE/2)):NEXT TE
330 GOTO 140
```

Tastiera

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: gioco di riflessi (imparare la tastiera).
Difficoltà del gioco: elevata.

IL GIOCO: Una riproduzione della tastiera del vostro calcolatore appare sullo schermo; un punto appare sotto il carattere che si deve battere sulla tastiera. Più si batte in fretta e più il punto si sposta rapidamente (il tempo trascorso dall'inizio appare in fine partita).

IL PROGRAMMA: Una variabile A$ dimensionata alla linea 60, conterrà la lista dei relativi tasti. I valori dei tasti sono inseriti dalla linea 70 alla linea 120 per chiamata di un sottoprogramma posto in 300.
La variabile VA$ contiene la totalità dei codici ATASCII che corrispondono ai tasti della tastiera.
I cicli chiusi J (130-170) e I (140-160) disegnano la tastiera.
La linea 180 definisce la posizione del punto stampato dalla linea 190.
Il tasto battuto è testato alla linee 210-230 e il punto è cancellato dalla linea 250.
Le visualizzazioni di fine sono realizzate alle linee 270-290.

PER GIOCARE: Si lancia il programma con RUN. Si dispone allora di qualche secondo per battere il tasto che occupa la colonna sopra la pallina, poi un altro punto è visualizzato, ecc.
La durata della partita non è limitata; basta battere 0 per ottenere il punteggio, ed il tempo totale.
Fare RUN per un'altra partita.

ESTENSIONI POSSIBILI: Si potrà pensare di fare lampeggiare le lettere, o di visualizzarle a colori, al posto di indicare la colonna con una pallina. Si potranno fare apparire le lettere giuste nella parte inferiore dello schermo ed eventualmente allenarsi a battere un testo che sfila in alto sullo schermo, lettera per lettera.

```
10 REM TASTIERA
40 GRAPHICS 2
50 T=0:S=0:P=0
60 DIM A$(30),S$(10),V$(20),VA$(60)
70 S$="QWERTYUIOP":V$="47464240454311130810":J=1
80 GOSUB 300
90 S$="ASDFGHJKL;":V$="63625856615701050002":J=2
100 GOSUB 300
110 S$="ZXCVBNM,./":V$="23221816213537323438":J=3
120 GOSUB 300
130 FOR J=1 TO 3
140 FOR I=1 TO 10
150 POSITION I+J+2,2*J:PRINT #6;A$(10*(J-1)+I,10*(J-1)+I)
160 NEXT I
170 NEXT J
180 X=INT(RND(1)*10+1):Y=INT(RND(1)*3+1)
190 POSITION X+Y+2,2*Y+1:PRINT #6;"*"
195 POKE 764,255:CLA1=PEEK(764)
200 FOR F=1 TO 100+RND(1)*80:CLA=PEEK(764)
205 T=T+1
210 IF CLA=CLA1 THEN NEXT F
220 IF CLA=VAL(VA$(20*(Y-1)+2*X-1,20*(Y-1)+2*X)) THEN S=S+1
230 IF CLA=50 THEN 270
240 P=P+1
250 POSITION X+Y+2,2*Y+1:PRINT #6;" "
260 GOTO 180
270 PRINT S;" Giuste su ";P;" prove."
275 PRINT "Il tuo tempo :";T
280 FOR T=1 TO 1000:NEXT T:GRAPHICS 0:END
300 A$(10*(J-1)+1)=S$
310 VA$(20*(J-1)+1)=V$:RETURN
```

LIVELLO 5

N.	Nome del programma	Tema e particolarità
1	Calcolatrice	Un programma di simulazione molto reale
2	Smokey	Fumate senza rischi con questo originale programma per 1 giocatore
3	Evasione	Ci vorrà molta fortuna per riuscirci
4	Corsa automobilistica	Un gioco di riflessi non molto difficile per 1 giocatore
5	Poker	Una base di partenza per mettere a punto il vostro programma
6	Scacchi	Un programma semplificato per 2 giocatori
7	Pericolo spaziale	Sarete abbastanza abili per evitare tutte le meteoriti?
8	Tic Tac Toe	Un classico ma non senza interesse (2 giocatori)
9	Dama	Per 2 giocatori: classico
10	Anagrammi	Un buon gioco di riflessi: per 2 giocatori
11	Uova	Non lasciate fare la gallina diabolica
12	Quattro a quattro	Un gioco di riflessione un po' difficile per 2 giocatori
13	"Puzzle" di lettere	Un proverbio è nascosto: a voi scoprirlo
14	Bersagli	Un po' di abilità e molta fortuna
15	Gocce	Una vera gag visiva, da esplorare
16	Ostacoli	Molti ostacoli da evitare in questo gioco di abilità per 1 giocatore
17	Catture nello spazio	La missione sarebbe facile senza queste maledette meteoriti
18	Anomalia	Esercitate la vostra perspicacia riconoscendo i simboli
19	Tanti in uno	Un terreno di gioco e le pedine, per 2 giocatori
20	Percorso del cavaliere	Molto spettacolare e istruttivo per più di un motivo
21	Muro di mattoni	Un classico in versione completa
22	Othello	Per 2 giocatori: gira le pedine per voi
23	"Puzzle" dispettoso	Un sottoprodotto del gioco di Sam Lloyd
24	Super scorrimento	Dopo le cifre pari e dispari perché non anche le lettere

Le tecniche particolari all'ATARI

IL RICONOSCIMENTO DEGLI OGGETTI SULLO SCHERMO: LETTURA DI UN CARATTERE POSTO ALLA LINEA Y, COLONNA X, CON LOCATE X,Y,C

L'ATARI dispone di un'istruzione specifica, molto utile per numerosi giochi, che permette di riconoscere il codice ASCII del carattere presente sullo schermo all'intersezione tra la colonna X e la linea Y: questo permetterà di identificare facilmente questo carattere; C contiene il codice ASCII del carattere presente.

Attenzione, questa lettura dello schermo non si mantiene, conviene riportare il valore che leggete con l'aiuto di una visualizzazione posizionata alle stesse coordinate dello schermo.

LE ISTRUZIONI POP e TRAP

Quando siete in un sottoprogramma (vedere livello 1) ne uscite sempre con RETURN. Può essere interessante uscire direttamente a un indirizzo dato.

L'istruzione POP permette di farlo indicando all'ATARI di accorgersi che il sottoprogramma è terminato. Nel caso contrario se il numero di livelli di sottoprogrammi è limitato, otterrete subito un superamento di capacità.

È intelligente imparare dagli errori per scrivere un programma. Così posizionare un punto a delle coordinate X,Y fuori dallo schermo provoca un errore e arresta il programma; l'istruzione TRAP permette di rivelare questo errore, senza che il calcolatore intervenga, riportando il programma all'indirizzo dove l'errore è stato generato.

Così TRAP 5000 rinvia, in caso d'errore, il programma all'istruzione 5000.

Per eliminare questa possibilità basta usare TRAP seguito da una cifra superiore a 32767.

IL JOYSTICK

È una delle attrattive dell'home computer ATARI, perché facilita enormemente gli spostamenti dei mobili sullo schermo.
Molti giochi di questo libro potranno essere modificati per una miglior utilizzazione, con l'aiuto di questo "manico di scopa": il programma 4 a 4 ne è l'illustrazione.

Calcolatrice

RIFERIMENTI: Programmazione: molto difficile.
Categoria del gioco: simulazione di una calcolatrice.
Difficoltà del gioco: media.

IL GIOCO: Si tratta della simulazione, con visualizzazione in grande formato, del funzionamento di una piccola calcolatrice tascabile, del tipo a 4 operazioni. Le applicazioni di questo programma sono evidenti e il suo funzionamento è semplice da controllare.

IL PROGRAMMA: Le variabili B$ e C$, delle linee 20 e 30, contengono le informazioni codificate che permetteranno il disegno delle cifre in grande formato.
Le linee 60-150 si occupano della gestione degli inserimenti. Il ciclo I(160-190) realizza la raffigurazione, in grande formato, delle cifre sullo schermo, man mano che gli input arrivano.
Il punto decimale ed il segno — sono gestiti dalle linee 200 e 210.
Il numero totale dei simboli inseriti è controllato dalla linea 220.
Il risultato è iscritto grazie ad un programma funzionante automaticamente che si trova alle linee 270-390 (l'automatismo è gestito dalla linea 230, nel programma principale).
I sottoprogrammi posti alle linee 400-410, 420-430, 440-450 e 460-470 effettuano l'operazione indicata, secondo il valore di Z (linea 310).
Ogni risultato è memorizzato in A (F-1) (linea 330).
La routine d'errore in 2000 evita l'arresto del programma quando VAL(R$) è utilizzata per caratteri che non siano cifre.

PER GIOCARE: Si dovrà lanciare il programma con RUN.
Poi, si battono le cifre del primo numero da inserire che appariranno una per una, man mano (non si potranno inserire più di 8 simboli, compreso il segno — ed il punto decimale). La battuta di un segno di operazione lascia la visualizzazione intatta, ma l'introduzione della prossima cifra o simbolo provoca la cancellazione immediata e la visualizzazione del nuovo simbolo. È da notare che le cifre si introducono da sinistra a destra, senso usuale di scrittura.
L'apparecchio accetta, come segni operatori i simboli +, —, *, /; il segno = provocherà l'apparizione del risultato. Le operazioni sono concatenate automaticamente, il risultato precedente è preso in considerazione come operatore del calcolo seguente.
Per fare un'altra operazione, bisognerà battere BREAK poi RUN.

ESTENSIONI POSSIBILI: Si dovrà, innanzitutto, migliorare la visualizzazione del segno — (raffigurato qui da un tratto), poi si potrà migliorare il disegno delle cifre (utilizzando per esempio, il generatore dell'ATARI) si può anche pensare ad un'estensione a più di 8 cifre, autorizzare delle operazioni più complesse (COS, SIN, ...), aggiungere delle memorie, ecc).

```
10 REM CALCOLATRICE
12 SETCOLOR 1,2,10:SETCOLOR 2,0,0
15 DIM A$(1),B$(50),C$(12),R$(20),A(10)
17 FOR I=1 TO 10:A(I)=0:NEXT I
20 C$="****    ** *":REM sostituire * con un bianco in colore inverso
30 B$="144413333313121131314413312131121411333314141141 31"
32 POKE 752,1
35 OPEN #1,4,0,"K:"
40 F=1:L=0:PRINT "}"
50 N=0:T=0:K=0:R$=""
60 GET #1,CLA:A$=CHR$(CLA)
70 IF K=1 THEN N=0:R$="0":K=0
80 IF A$="=" THEN 260
90 X=(1*A$="+")+(2*A$="-")+(3*A$="*")+(4*A$="/")
100 R$(LEN(R$)+1)=A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$<"." OR A$>"9" OR A$="/" THEN 60
140 IF N=0 THEN PRINT "}"
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(B$(I+5*VAL(A$),I+5*VAL(A$)))-2
180 POSITION N,I+3:PRINT C$(S,S+2)
190 NEXT I
200 IF A$="-" THEN PRINT "}":POSITION 1,7:PRINT CHR$(160);CHR$(160)
210 IF A$="." THEN POSITION N+1,8:PRINT CHR$(160)
220 N=N+4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THEN 380
240 IF K=1 AND L<>1 THEN 50
250 T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 GOSUB 2000:IF (A$="*" OR A$="/") AND VA=0 THEN A(F)=1:GOTO 300
290 GOSUB 2000:A(F)=VA
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>"=" THEN F=F+1
325 IF A$="=" THEN F=2
330 GOSUB 2000:A(F-1)=VA
340 GOSUB 2000:IF VA=0 THEN A(F-1)=A(F)
350 GOSUB 2000:R$=STR$(VA)
370 T=1
380 A$=R$(T,T)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(F)=VAL(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(F)=VAL(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(F)=VAL(R$):L=1
470 RETURN
2000 TRAP 2100
2010 VA=VAL(R$)
2020 RETURN
2100 VA=0
2110 TRAP 50000
2120 RETURN
```

Smokey

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Una colonna di fumo grigio appare sullo schermo. Il giocatore, rappresentato da un punto, diventa visibile sulla destra. In un primo tempo deve posizionarsi sulla colonna di fumo, poi, partendo da lì, ingoierà automaticamente il fumo, a patto che riesca a mantenersi sulla traccia.

IL PROGRAMMA: La linea 90 fa apparire il fumo nel basso dello schermo; la linea 100 permette di cambiare (verso destra o verso sinistra) la direzione della colonna, e la linea 120 fa salire la colonna.
Il giocatore appare grazie alla linea 140, ed il suo spostamento è effettuato dalle linee 180 e 190.
Le linee 250 e 260 si incaricano della visualizzazione del risultato.

PER GIOCARE: Si deve lanciare l'esecuzione del programma con RUN.
Il fumo comincia ad alzarsi, mentre il giocatore apparso nella forma di un cerchio colorato in un angolo dello schermo si dovrà spostare con l'aiuto delle tre frecce. Appena arriva sul fumo, comincerà automaticamente ad ingoiarlo, e dovrà cercare di seguire la colonna senza deviare (grazie alle frecce). Vince se arriva nella parte inferiore dello schermo prima che la colonna arrivi in alto.
Il messaggio finale indica allora BRAVO o PERSO, così come il punteggio ottenuto.

ESTENSIONI POSSIBILI: Si potrà, conservando l'idea di partenza, imporre delle altre regole: per esempio ingoiare tutto il fumo, che potrà anche ripartire da un altro posto se si devia la colonna iniziale.
Si potrà anche migliorare i controlli di fine partita e la qualità della grafica.

```
10 REM SMOKEY
20 GRAPHICS 0:POKE 752,1
30 SETCOLOR 1,3,10:SETCOLOR 2,1,2:SETCOLOR 2,1,2
60 S=0:X=23:Y=INT(RND(1)*23)
70 U=0:V=15
71 U=1
80 PRINT "}"
90 POSITION Y,X:PRINT CHR$(0)
100 Y=Y+(RND(1)<0.5 AND Y<23)-(RND(1)<0.5 AND Y>0)
110 T=1
120 X=X-T
130 IF X=0 THEN 250
140 POSITION V,U:PRINT CHR$(20)
145 POKE 764,255
150 IF T<>0 THEN FOR I=1 TO 300:CLA=PEEK(764):IF CLA<>255 THEN 170
160 IF T<>0 THEN NEXT I
170 POKE 764,225:POSITION V,U:PRINT " "
180 U=U+(CLA=15)*(T=1)*(U<23)
190 U=U+(T=0)
200 V=V+(CLA=7)*(V<24)-(CLA=6)*(V>0)
210 LOCATE V,U,C:POSITION V,U:PRINT CHR$(C);
220 IF C<>0 THEN 90
230 T=0:S=S+1
240 IF U<22 THEN 140
250 IF T=0 THEN PRINT "BRAVO":FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR T=1 TO 10
:NEXT T:NEXT I:GOTO 270
260 PRINT "PERSO!...":FOR I=1 TO 3:SOUND 0,63,10,4:FOR T=1 TO 5:SOUND 0,0,0,0:NE
XT T:NEXT I
270 PRINT "Punteggio :";S
1000 DATA 63,47,40
```

Evasione

RIFERIMENTI: Programmazione: facile.
Categoria del gioco: solitario.
Difficoltà del gioco: bassa.

IL GIOCO: Il vostro solo problema: riuscire ad attraversare lo schermo, senza che l'ATARI vi abbia prima colpito con uno dei suoi numerosi proiettili esplosivi che vi sparerà addosso.
Potete spostarvi in 4 direzioni con l'aiuto di 4 frecce e dovete riuscire ad uscire dal terreno dal lafo destro, senza, naturalmente, essere mai stati colpiti.

IL PROGRAMMA: La linea 50 definisce la posizione di partenza del giocatore.
La linea 60 sceglie la coordinata orizzontale della casella dove l'ATARI sparerà (legata alla vostra posizione). L'utilizzo di TRAP evita l'errore di uno sparo fuori dal territorio autorizzato provocando la ricerca di un nuovo sparo e la linea 50 sceglie la coordinata verticale (arbitraria) di questa casella.
Lo spostamento del giocatore è controllato dalle linee 140 e 150. La sua traiettoria è tracciata dalla linea 120 e la sua posizione attuale (sullo schermo una freccia) è stampata dalla linea 160.
La linea 170 controlla se è stato toccato o no.
La linea 180 effettua il controllo di fine partita.

PER GIOCARE: Si comincia l'esecuzione del programma con RUN.
Un primo sparo ha luogo, dopo che il giocatore appare, rappresentato da una freccia.
I 4 tasti di direzione servono a spostare il giocatore.
Se è colpito, si ottiene, sullo schermo, il messaggio PERSO e la partita si ferma.
Se riesce ad uscire alla destra dello schermo senza essere colpito, si ottiene il messaggio BRAVO, e la partita è ugualmente terminata.
Si può terminare in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Il colore ed una grafica più curata miglioreranno considerevolmente l'attrattiva del gioco. Si potrà anche rendere il lancio delle bombe più casuale, introdurre degli ostacoli da aggirare, aggiungere dei controlli, fare partecipare vari giocatori, ecc.

```
10 REM EVASIONE
20 MUSIQUE=500:BEEP=600:SISODO=1000:DOMISO=2000
40 GRAPHICS 0:SETCOLOR 1,6,6:SETCOLOR 2,13,10
50 X=0:Y=12
60 H=X+2-INT(RND(1)*6)
70 K=INT(23*RND(1))
80 IF X<3 THEN 120
100 POSITION H,K:PRINT CHR$(160)
120 GOSUB BEEP:GOSUB BEEP:POSITION X,Y:PRINT "."
130 P=PEEK(764):POKE 764,255
140 X=X-(P=6)+(P=7)
150 Y=Y-(P=14)+(P=15)
160 POSITION X,Y:PRINT CHR$(123)
170 IF X>3 AND (ABS(X-H)+ABS(Y-K)<=1) THEN PRINT "PERSO":RESTORE SISODO:GOSUB MU
SIQUE:END
180 IF X<38 THEN 60
190 PRINT "BRAVO!":RESTORE DOMISO:GOSUB MUSIQUE:END
500 FOR I=1 TO 3
510 READ VA:SOUND 0,VA,10,4:FOR T=1 TO 15:NEXT T:SOUND 0,0,0,0:FOR T=1 TO 5:NEXT
 T
520 NEXT I
530 RETURN
600 SOUND 0,31,10,10:FOR T=1 TO 5:NEXT T:SOUND 0,0,0,0:RETURN
1000 DATA 64,81,126
2000 DATA 126,96,81
```

Corsa automobilistica

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco di riflessi (un giocatore).
Difficoltà del gioco: media.

IL GIOCO: Guidate su una strada molto sinuosa, senza nessun ostacolo, e dovete cercare di seguirla per il maggior tempo possibile. La traccia del vostro percorso resta visualizzata in permanenza, e il vostro punteggio appare subito dopo un incidente.

IL PROGRAMMA: La linea 20 simula il rumore di un motore.
La stampa della strada è assicurata dal ciclo I (70-100) e la linea 90 traccia i bordi della strada.
Il ciclo fittizio N permette di far avanzare la strada verso la macchina.
La linea 110 fa avanzare la strada, e la linea 120 stampa la macchina.
La linea 150 controlla il suo movimento, e la linea 160 determina se la strada fa o no delle curve.
La linea 170 guarda se abbiamo toccato il bordo della strada.
La linea 175 controlla che non si esca dalla strada.
La linea 180 incrementa il punteggio e la linea 190 cancella la vecchia posizione della macchina.
La linea 220 visualizza il tempo fino all'incidente.

PER GIOCARE: Lanciare l'esecuzione del programma con RUN, il disegno della strada comincia ad evolversi, ed appare il vostro veicolo, rappresentato da una macchina.
Per spostarlo verso destra o verso sinistra, dovrete utilizzare i tasti → e ← (frecce di spostamento del cursore). Dovrete evitare per il maggior tempo possibile di toccare il bordo nero, il che provocherà l'arresto del programma e la visualizzazione del messaggio INCIDENTE, seguito dall'indicazione del punteggio realizzato.
Si potrà fermare lo svolgimento della partita in ogni istante battendo BREAK.

ESTENSIONI POSSIBILI: Si può immaginare di giocare in molti, ognuno al suo turno, registrando il miglior punteggio.
Si può anche pensare di allargare la pista, e di fare correre varie macchine. Potremo anche ispirarci ai giochi dei bar che sono particolarmente numerosi su questo tema.

```
10 REM CORSA AUTOMOBILISTICA
20 SOUND 2,123,8,2
50 T=0:GRAPHICS 0:POKE 752,1
60 X=23:Y=11:A=5:B=10:S=0
70 FOR I=0 TO 6
80 POSITION Y+I,X:PRINT CHR$(160);
90 IF I<2 OR I>4 THEN POSITION Y+I,X:PRINT CHR$(174);
100 NEXT I
110 PRINT
120 POSITION B,A:PRINT CHR$(16);:POSITION C,A-1:PRINT CHR$(32+128*(N=18));
130 FOR I=1 TO 10:CLA=PEEK(764):T=T+0.1:IF CLA=255 THEN NEXT I
140 C=B:POKE 764,255
150 B=B-(CLA=6)+(CLA=7)
160 Y=Y+(2*RND(1)*(Y<25))-(2*RND(1)*(Y>3))
165 IF N<18 THEN N=N+1:GOTO 70
170 LOCATE B,A,EC:POSITION B,A:PRINT CHR$(EC);:IF EC=32 THEN 210
175 LOCATE B,A+1,EC:POSITION B,A+1:PRINT CHR$(EC);:IF EC=32 THEN 210
180 S=S+1
190 REM POSITION C,A:PRINT CHR$(160);
200 GOTO 70
210 PRINT "INCIDENTE :punteggio :";S
220 PRINT :PRINT "TEMPO :";INT(T):FOR TE=1 TO 255:SOUND 0,TE,10,4:NEXT TE:END
```

Poker

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco contro l'ATARI.
Difficoltà del gioco: bassa.

IL GIOCO: È una simulazione di una partita di poker tradizionale, effettuata con un mazzo di 32 carte, nella quale il giocatore avrà, come avversario, l'ATARI. Il giocatore riceverà 5 carte, individuate con dei numeri da 1 a 5, e potrà decidere di cambiarne alcune annunciando quelle che scarta. Le sue nuove carte saranno allora visualizzate, ed è in seguito il turno di gioco dell'ATARI.

IL PROGRAMMA: Le linee 50 e 60, definiscono le carte. Una variabile N è dimensionata a 16, alla linea 40, per ricevere 10 carte giocatore e 5 carte ATARI.
La presa sarà effettuata da un ciclo I (80-140) ed il controllo da un ciclo J (110-130).
La visualizzazione delle carte è realizzata dal sottoprogamma 500-610.
Un altro ciclo I (190-210) realizza la seconda presa del giocatore, la linea 240 permette la seconda visualizzazione, e il ciclo I (500-570) in cui si entra direttamente dopo, realizzerà la visualizzazione del gioco dell'ATARI.
La linea 590 permette di uscire direttamente dal sottoprogramma 580 senza passare da RETURN.

PER GIOCARE: L'ordine RUN lancia l'esecuzione del programma, e appaiono le 5 prime carte del giocatore.
Alla domanda CARTE RIFIUTATE?, il giocatore può rispondere 0 seguito da RETURN se vuole fermare la partita. Per giocare, risponderà semplicemente RETURN se si stima servito, oppure inserirà i numeri delle carte che rifiuta seguiti da RETURN.
Per esempio, se batte 235 e RETURN, il suo nuovo gioco conserverà le carte 1 e 4. Le sue carte saranno allora visualizzate così come quelle dell'ATARI, quest'ultimo non avendo, in questo programma, la possibilità di cambiare delle carte.
Basterà in seguito battere un tasto qualunque per cominciare un'altra mano.

ESTENSIONI POSSIBILI: Saranno naturalmente numerose: da una visualizzazione grafica a colori delle carte fino al miglioramento del gioco dell'ATARI, l'introduzione delle regole sulle scommesse ed anche, perché no, la possibilità di bluff.

```
10 REM POKER
20 GRAPHICS 0
30 SETCOLOR 1,5,6:SETCOLOR 2,12,2
40 DIM N(15),R$(6),V$(24),C$(8)
50 V$="7 8 9 10FADOREAS"
60 C$="FIQUCUPI"
70 PRINT "}"
80 FOR I=1 TO 15
90 N(I)=INT(RND(1)*32+1)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 POSITION 0,12:PRINT "CARTE RIFIUTATE:"
170 GOSUB 500
180 IF R$="" THEN 220
190 FOR I=1 TO LEN(R$)
200 N(VAL(R$(I,I)))=N(I+5)
210 NEXT I
220 PRINT "}"
230 P=1
240 GOSUB 500
250 K=K+10
500 FOR I=1 TO 5
510 J=1+INT((N(I+K)-1)/8)
520 F=N(I+K)-8*(INT((N(I+K)-1)/8))
530 R$=V$(2*F-1,2*F)
540 POSITION 7*I-2,K:PRINT I
550 POSITION 7*I-4,K+2:PRINT R$;":";C$(2*J-1,2*J)
560 POSITION 7*I-4,K+3:PRINT "-----"
570 NEXT I
580 IF P=1 THEN P=0:RETURN
590 IF K=10 THEN PRINT :PRINT "Battere <RETURN> per continuare";:INPUT R$:POP :G
OTO 70
600 POSITION 18,12:INPUT R$
610 IF R$="0" THEN END
620 RETURN
```

Scacchi

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco di riflessione, per due giocatori.
Difficoltà del gioco: elevata.

IL GIOCO: Il programma propone solamente ai due giocatori di fare una partita servendosi dello schermo come scacchiera. Non c'è controllo della validità delle mosse, ma solamente la gestione della posizione degli scacchi sulla scacchiera. Le pedine e gli scacchi sono disegnati con l'aiuto delle istruzioni di lettere e cifre. Il colore permette di distinguere ogni giocatore.

IL PROGRAMMA: Le linee 100-200 disegnano la scacchiera. Le linee 210 e 220 definiscono le condizioni iniziali, con l'aiuto di un array: R(I,J), per i codici degli scacchi, e C (I, J) per il colore.
I cicli chiusi I (240-290) e J (250-280) disegnano gli scacchi.
La linea 320 permette l'inserimento della linea e della colonna di partenza, con controllo alla linea 350.
La linea 370 permette d'inserire colonna e linea d'arrivo.
La linea 400 annota la nuova posizione.
La linea 410 assicura il cambio del giocatore.

PER GIOCARE: Si lancia l'esecuzione con RUN.
Viene disegnata la scacchiera poi i pezzi, ed infine la scritta giocatore 1 seguito da Partenza (C/L). Si inseriscono la colonna e la linea di partenza nella maniera E7 per esempio, si ottiene allora la scritta Arrivo (C/L) e si risponde nello stesso modo. Lo spostamento viene allora effettuato, e sarà il turno dell'altro giocatore.
Per finire, basta battere BREAK.

ESTENSIONI POSSIBILI: Si potrà cercare di realizzare una grafica più accurata, introdurre i controlli di movimento dei pezzi. E, perché no, scrivere un programma che permetta all'ATARI di comportarsi da buon partner. Buon lavoro!

```
10 REM SCACCHI
20 DIM R(8,8),C(8,8)
30 DIM A$(3),B$(3),C$(3),D$(2),V$(7),PI$(1)
35 V$="CRTQPF "
40 A$="":REM battere 3 volte <CONTROL>U
60 B$="   ":REM battere 3 bianchi in inverse
70 C$="":REM battere in inverse 3 volte <CONTROL>U
90 GRAPHICS 0:SETCOLOR 1,3,2:SETCOLOR 2,13,6
95 POKE 752,1
100 PRINT "  A  B  C  D  E  F  G  H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J)/2=INT((I+J)/2) THEN 150
140 POSITION 3*J,2*I-1:PRINT A$:PRINT I:POSITION 3*J,2*I:PRINT B$:POSITION 3*J,2
*I+1:PRINT C$
150 IF I=2 OR I=7 THEN R(I,J)=4
160 IF I<>2 AND I<>7 THEN R(I,J)=6
170 C(I,J)=1
180 IF I=7 OR I=8 THEN C(I,J)=6
190 NEXT J
200 NEXT I
210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:R(1,5)=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
230 H=1
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I
263 IF R(I,J)=6 THEN POSITION X,Y:PRINT CHR$(160-128*((I+J)/2=INT((I+J)/2))):GOT
O 280
265 PI$=V$(R(I,J)+1,R(I,J)+1)
267 PI$=CHR$(ASC(PI$)+128*C(I,J))
270 POSITION X,Y:PRINT PI$
280 NEXT I
290 NEXT J
300 POSITION 28,6:PRINT "           ":POSITION 14,20:PRINT "      ":POSITION 14,22:
PRINT "      "
310 POSITION 0,18:PRINT "GIOCATORE ";2-H
320 POSITION 0,20:PRINT "PARTENZA (C/L):";:INPUT D$
330 POSITION 28,6:PRINT D$
340 GOSUB 500
350 IF R(L,C)=6 OR (C(L,C)=1 AND H=1) OR (C(L,C)=6 AND H=0) THEN 300
360 T=R(L,C):S=C(L,C):R(L,C)=6
370 POSITION 0,22:PRINT "ARRIVO (C/L):";:INPUT D$
380 POSITION 31,6:PRINT "- ";D$
390 GOSUB 500
400 R(L,C)=T:C(L,C)=S
410 H=ABS(H-1)
420 GOTO 240
500 L=VAL(D$(2,2)):C=ASC(D$(1,1))-64
510 RETURN
```

Pericolo spaziale

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco di riflessi, per un giocatore.
Difficoltà del gioco: media.

IL GIOCO: Siete ai comandi di una nave spaziale, che è appena penetrata in uno sciame di meteoriti. Dovete cercare di uscire dallo sciame senza urtare nessuna meteorite. Il vostro punteggio sarà visualizzato alla fine della partita, quando urterete una meteorite.

IL PROGRAMMA: La posizione di ogni meteorite è determinata dalle linee 60 e 80 e sono stampate dalle linee 70 e 90.
L'avanzamento delle meteoriti è realizzato dalla linea 100.
Le linee 100 e 110 verificano se avete o no urtato una meteorite.
La linea 130 disegna la vostra astronave, e la linea 160 permette di spostarla.
Il contatore si trova alla linea 170.
La linea 200 simula un'esplosione.

PER GIOCARE: Il gioco è lanciato con l'ordine RUN.
La vostra astronave appare nel mezzo dello schermo, e le meteoriti vi vengono incontro partendo dalla parte inferiore dello schermo. Per spostarvi, potete utilizzare i tasti di spostamento orizzontale (← →).
Si può fermare la partita in ogni istante battendo BREAK. Altrimenti, il gioco si ferma quando urtate una meteorite, e appare il punteggio.

ESTENSIONI POSSIBILI: Sarebbe interessante rendere il gioco sempre più difficile, facendo apparire un numero sempre maggiore di meteoriti man mano che la partita si svolge.
Si può anche cercare di distruggere le meteoriti polverizzandole man mano che arrivano.

```
10 REM PERICOLO SPAZIALE
40 GRAPHICS 0:POKE 752,1
45 SETCOLOR 1,2,12:SETCOLOR 2,5,2
50 X=18:Z=0:Y=X
60 H=INT(RND(1)*39)
70 IF RND(1)<0.7 THEN POSITION H,23:PRINT CHR$(160);
80 K=INT(RND(1)*40)
90 IF RND(1)<0.6 THEN POSITION K,23:PRINT CHR$(160);
100 PRINT :LOCATE X,12,EB:POSITION X,12:PRINT CHR$(EB);:LOCATE X+1,12,EC:POSITIO
N X+1,12:PRINT CHR$(EC);
110 IF EB=160 OR EC=160 THEN 190
120 POSITION Y-1,11:PRINT "    ":POSITION Y-1,10:PRINT "    "
130 POSITION X,12:PRINT "##"
140 CLA=PEEK(764):POKE 764,255
150 Y=X
160 X=X+(CLA=7)*(X<38)-(CLA=6)*(X>0)
170 Z=Z+1
180 GOTO 60
190 POSITION 0,23:PRINT :PRINT "     Il punteggio e' di ";Z;" punti."
200 FOR I=1 TO 10:SOUND 0,145,4,I:FOR T=1 TO 15:NEXT T:SETCOLOR 2,5,14*(I/2=INT(
I/2)):NEXT I
```

Tic Tac Toe

RIFERIMENTI: Programmazione: media.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: bassa.

IL GIOCO: Un grande classico dei giochi scolastici, proposto qui in una versione che non comporta il gioco da parte del calcolatore; ogni giocatore dovrà posizionare nel suo turno uno 0 o una X in una casella libera della griglia di gioco, e il primo che riesce ad allineare tre simboli identici ha vinto (l'allineamento può essere orizzontale, verticale, o diagonale).

IL PROGRAMMA: La linea 25 autorizza la visualizzazione con funzione GRAPHICS 2 sotto forma di caratteri grafici e di minuscole.
La griglia di gioco è disegnata con l'aiuto di un ciclo I (40-65) che legge i DATA.
La linea 100 si occupa di visualizzare il simbolo corrispondente al giocatore che ha la mano.
La linea 130 fa lampeggiare il simbolo contenuto nella casella; le linee 160 e 170 permettono di spostare il simbolo nella griglia.
La linea 180 verifica se la casella dove si vuole giocare è libera e se si desidera giocare.
La linea 190 registra il colpo, e la linea 200 realizza il cambio del giocatore.

PER GIOCARE: La griglia di gioco è visualizzata con l'ordine RUN.
Un simbolo lampeggiante appare allora in alto e a sinistra dello schermo; il giocatore può spostarlo nella griglia con l'aiuto delle quattro frecce (←, →, ↑, ↓).
Quando arriva sulla casella dove si desidera giocare, basta che prema il tasto 0, la mossa sarà rifiutata se la casella non è libera.
Se la casella è libera, il simbolo corrispondente a questo giocatore sarà iscritto, e si cambia di giocatore. Quando la partita è finita, conviene battere BREAK per uscire dal programma. Si potrà in seguito fare un'altra partita con RUN.

ESTENSIONI POSSIBILI: Si potrebbe realizzare dopo ogni colpo un controllo per verificare se uno dei giocatori ha vinto. Si può anche realizzare un gioco per il calcolatore ma bisognerebbe evitare che sia troppo sistematico (per esempio, non giocare sempre la prima mossa sulla casella centrale).

```
10 REM TIC TAC TOE
15 HAUT=1000:INTER=1010:MILIEU=1020:BAS=1030
20 GRAPHICS 2
22 SETCOLOR 0,0,0
25 POKE 756,226
30 DIM C$(7)
40 RESTORE HAUT:READ C$:POSITION 6,1:PRINT #6;C$
45 FOR I=1 TO 6
50 RESTORE INTER:IF I=2*(INT(I/2)) THEN RESTORE MILIEU
55 READ C$:POSITION 6,1+I:PRINT #6;C$
60 NEXT I
65 READ C$:POSITION 6,7:PRINT #6;C$
100 PRINT "}":PRINT "Giocatore ";CHR$(79+K*9);
110 I=1:J=1
120 LOCATE 5+2*I,2*J,C
130 FOR N=1 TO 2:POSITION 5+2*I,2*J:PRINT #6;CHR$(C+128*(SGN(128-C))*(N=1)):NEXT
 N
150 CLA=PEEK(764)
155 POKE 764,255
160 I=I-(CLA=6 AND I>1)+(CLA=7 AND I<3)
170 J=J-(CLA=14 AND J>1)+(CLA=15 AND J<3)
180 IF C<>32 OR CLA<>50 THEN 120
190 POSITION 5+2*I,2*J:PRINT #6;CHR$(79+K*9+128)
200 K=ABS(K-1)
205 VA=63
210 IF K=1 THEN VA=126
215 FOR T=1 TO 10:SOUND 0,VA,10,15:NEXT T:SOUND 0,0,0,0
220 GOTO 100
1000 DATA qrwrwre:.battere simultaneamente <CONTROL>
1010 DATA | | | |
1020 DATA arsrsrd:.vedere 1000
1030 DATA zrxrxrc:.vedere 1000
```

Dama

RIFERIMENTI:

Programmazione: molto difficile.
Categoria del gioco: due giocatori.
Difficoltà del gioco: elevata.

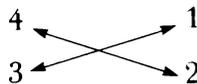
IL GIOCO:

È il gioco classico della dama alla francese, ma un po' incompleto visto che non prevede la possibilità di fare delle dame. La regola del gioco resta classica, ma i controlli sono stati un po' semplificati. I movimenti delle pedine sono inseriti in una sola manovra (posizione di partenza e direzione dello spostamento) e le prese sono realizzate automaticamente, se sono possibili.
Invece il cambio del giocatore è realizzato ad ogni mossa, quindi bisognerà eventualmente battere 0 per poter rigiocare se c'era stato un salto o se il colpo non era possibile.

IL PROGRAMMA:

I cicli I (80-170) e J (90-160) disegnano la scacchiera e definiscono le posizioni iniziali delle pedine nere e bianche. Le posizioni sono in un array R (X,Y) e il colore in un altro array C (X,Y).
La stampa delle pedine vera e propria è effettuata dai cicli chiusi J (190-250) e I (200-240). La linea 270 indica il giocatore di turno.
La mossa è inserita alla linea 270.
La linea 280 realizza il cambio del giocatore (si batte 0).
La linea 300 verifica se lo spostamento è possibile.
La linea 330 tratta il caso di uno spostamento e la linea 340 quello di un salto.
La linea 380 assicura il cambio del giocatore.

PER GIOCARE:

Dopo il lancio del programma con RUN la scacchiera è visualizzata con le sue pedine e un cerchio colorato appare sotto per indicare il giocatore che deve iniziare.
Si gioca indicando con una parola di 3 lettere (che può cominciare con uno 0) la linea e la colonna della pedina da spostare, poi la direzione dello spostamento (codificata da 1 a 4) secondo la convenzione

Per esempio 9D1 significa che la pedina bianca situata alla linea 9, colonna D, si sposta nella direzione 1 (non dimenticare di battere RETURN).

Allo stesso modo, 0G3 significherà pedina nera, linea 0, colonna G, direzione 3.
Il cambiamento del giocatore è effettuato automaticamente, bisognerà battere 0 per rigiocare se il colpo è stato rifiutato o se c'è stata una presa (terminare sempre con RETURN).
Il programma non controlla l'arrivo a dama e non proibisce gli spostamenti indietro.

ESTENSIONI POSSIBILI: Sarà abbastanza facile cominciare col migliorare i controlli (cambio di giocatore, mossa indietro che non corrisponderebbe a una presa, uscita dalla scacchiera). Si potrà pensare alla creazione di dame, al loro spostamento, ed eventualmente scrivere un programma che permetta all'ATARI di giocare con voi, con buone possibilità di vincita.

```
10 REM DAMA
20 DIM R(10,10),C(10,10)
25 DIM A$(3),B$(3),C$(3),D$(5)
30 A$="uuu":REM battere 3 volte <CONTROL> u
40 B$="   ":REM battere in inverse 3 bianchi
50 C$="uuu":REM battere 3 volte <CONTROL> u
60 GRAPHICS 0:POKE 752,1
70 PRINT "  A  B  C  D  E  F  G  H  I  J"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J)/2=INT((I+J)/2) THEN 120
110 POSITION 3*J,2*I-1:PRINT A$:PRINT I-1:POSITION 3*J,2*I:PRINT B$:POSITION 3*J
,2*I+1:PRINT C$
120 R(I,J)=0:C(I,J)=0
130 IF (I+J)/2<>INT((I+J)/2) THEN 160
140 IF J<5 THEN R(I,J)=1:C(I,J)=4
150 IF J>6 THEN R(I,J)=1:C(I,J)=1
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J)/2<>INT((I+J)/2) THEN 240
220 X=3*J+1:Y=2*I
230 POSITION X,Y:PRINT CHR$(32*(R(I,J)=0)+123*R(I,J)*(C(I,J)=4))
240 NEXT I
250 NEXT J
260 POSITION 24,22:PRINT "      "
270 POSITION 0,22:PRINT "Giocatore ";CHR$(123*(H=1));" ; la tua mossa :";:INPUT
D$
280 IF D$="0" THEN H=ABS(H-1):GOTO 260
290 GOSUB 400
300 IF R(L,C)=0 OR (C(L,C)=1 AND H=1) OR (C(L,C)=4 AND H=0) THEN 260
310 T=(D=1 OR D=2)-(D=4 OR D=3)
320 S=(D=3 OR D=2)-(D=1 OR D=4)
330 IF R(L+S,C+T)=0 THEN R(L,C)=0:R(L+S,C+T)=1:C(L+S,C+T)=1+3*(H=1):GOTO 380
340 IF C(L+S,C+T)=1+3*(H=0) AND R(L+S,C+T)=1 AND R(L+2*S,C+2*T)=0 THEN 360
350 GOTO 260
360 R(L,C)=0:R(L+S,C+T)=0:R(L+2*S,C+2*T)=1:C(L+2*S,C+2*T)=1+3*(H=1)
370 GOTO 190
380 H=ABS(H-1)
390 GOTO 190
400 L=VAL(D$(1,1))+1:C=ASC(D$(2,2))-64:D=VAL(D$(3,3))
410 RETURN
```

Anagrammi

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: gioco di riflessi (2 giocatori).
Difficoltà del gioco: secondo il livello.

IL GIOCO: Delle lettere sono scelte a caso dall'ATARI, il numero essendo in funzione del livello di difficoltà desiderato. Poi le lettere della parola così ottenuta sono mescolate e visualizzate nella parte inferiore dello schermo; una nuova parola è allora creata, in modo casuale, con le stesse lettere, ed è visualizzata sotto la prima e così via; i due giocatori (destro e sinistro) dispongono ognuno di un tasto (0 o 1) e devono premerlo appena vedono visualizzarsi in basso una parola già apparsa; il loro punteggio è allora calcolato ed apparirà un messaggio corrispondente al loro risultato.

IL PROGRAMMA: Il livello di gioco è inserito alla linea 50. Il ciclo I (80-100) realizza la scelta delle lettere.
Un secondo ciclo I (120-190) effettua la mescola, e la linea 140 permette di rivelare la pressione di un tasto.
La linea 200 visualizza una nuova parola, e fa salire di una linea sullo schermo le parole già ottenute.
La linea 240 cerca quale dei due giocatori ha premuto il tasto.
Il ciclo I (250-320) esplora le parole presenti sullo schermo linea per linea, mentre il ciclo J (270-300) prende una per una le lettere di queste parole.
Si ottiene così una nuova parola che è paragonata alla parola iniziale alla linea 310 ed il risultato BRAVO appare se è esatto.
Altrimenti, la partita si ferma sul messaggio PERSO (linea 330).

PER GIOCARE: Si lancia il programma con RUN.
Si dovrà anche inserire il livello di difficoltà (in teoria da 1 a 3) e fare RETURN. Le parole si visualizzano partendo dal basso dello schermo, e salgono man mano; se uno dei giocatori crede di riconoscere nella parola apparsa una delle parole già visualizzate, dovrà immediatamente premere il suo tasto (1 o 0), ed il suo risultato sarà visualizzato dopo un istante.
Allora la partita sarà terminata.

ESTENSIONI POSSIBILI: Si potranno utilizzare non solo le parole, modificare i tempi di visualizzazione, fare partecipare più di due giocatori, introdurre il suono ed il colore, ecc.

```
10 REM ANAGRAMMI
30 DIM A$(6),B$(6),H$(16),R$(1),S$(6)
40 GRAPHICS 0:POKE 752,1
50 PRINT "Livello (da 1 a 3) :";:INPUT N
60 IF N<1 OR N>3 THEN 40
70 S$=""
80 FOR I=1 TO N+3
90 S$(LEN(S$)+1)=CHR$(INT(RND(1)*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND(1)*LEN(S$)+1)
140 CLA=PEEK(764):POKE 764,255
150 IF CLA=31 OR CLA=50 THEN 240
160 IF B$(N,N)="-" THEN 130
170 A$(LEN(A$)+1)=B$(N,N)
180 B$(N,N)="-"
190 NEXT I
200 POSITION 0,23:PRINT A$
210 FOR I=1 TO 20*LEN(B$):NEXT I
220 S$=A$
230 GOTO 110
240 H$="Giocatore di destra"
245 IF CLA=31 THEN H$="Giocatore di sinistra"
250 FOR I=0 TO 21
260 B$=""
270 FOR J=0 TO LEN(S$)-1
280 LOCATE J,I,C:POSITION J,I:R$=CHR$(C):PRINT R$;
290 B$(LEN(B$)+1)=R$
300 NEXT J
310 IF B$=S$ THEN POSITION LEN(S$)+4,I:PRINT "BRAVO! ";H$:RESTORE 1000:GOSUB 500
:END
320 NEXT I
330 POSITION LEN(S$)+4,23:PRINT "Perso! ";H$;:FOR I=0 TO 5:SOUND 0,63,10,4*(I/2=
INT(I/2)):END
500 FOR I=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:FO
R TE=1 TO 3:NEXT TE:NEXT I:RETURN
1000 DATA 63,47,40
```

Uova

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Un uccello misterioso si sposta battendo le ali a caso sullo schermo. Depone delle uova (raffigurate da delle O) che voi (un quadrato colorato) dovrete cercare di recuperare. Il vostro punteggio (numero di uova recuperate) sarà visualizzato in permanenza; la partita si svolge in un tempo determinato.

IL PROGRAMMA: L'uccello è disegnato in due parti (per il movimento) alle linee 70 e 110; il suo spostamento casuale è generato dal sottoprogramma 210-300 (in particolare le linee 260-290).
La linea 130 permette di riciclare il programma sulla linea 70.
Lo spostamento del giocatore è gestito dalle linee 150 e 160, e il test che permette di rilevare la presenza di un uovo è effettuato dalla linea 170.
La linea 190 permette anche la visualizzazione del punteggio.
La fine del gioco è effettuata alla linea 230.

PER GIOCARE: Si lancerà il programma con RUN.
L'uccello comincia allora a spostarsi a caso sullo schermo e si mette a deporre le sue uova. Quando il simbolo che rappresenta il giocatore (un quadrato colorato) appare sullo schermo, costui può spostarsi utilizzando i 4 tasti direzionali. Si deve allora cercare di recuperare un massimo di uova prima della fine della partita.
Si potrà fermare il gioco in qualsiasi momento battendo BREAK.

ESTENSIONI POSSIBILI: Sullo stesso principio, potremo facilmente immaginare degli altri temi di gioco. Potremo naturalmente migliorare il colore, i suoni, la grafica, ed eventualmente, definire un gioco con vari giocatori, ognuno che cerchi di recuperare un massimo di uova.

```
10 REM UOVA
50 GRAPHICS 0:POKE 752,1:P=200:R=18:S=10:T=0
55 SETCOLOR 1,1,10:SETCOLOR 2,2,2:SETCOLOR 4,2,2
60 X=INT(RND(1)*36):Y=INT(RND(1)*21)
70 POSITION X,Y:PRINT CHR$(7);CHR$(14);CHR$(6)
80 CLA1=PEEK(764):FOR I=1 TO 30:CLA=PEEK(764)
90 IF CLA=CLA1 THEN NEXT I
95 GOSUB 140
97 POKE 764,255
100 GOSUB 210
110 POSITION X,Y:PRINT CHR$(6);CHR$(13);CHR$(7)
120 GOSUB 210
130 GOTO 70
140 POSITION R,S:PRINT " "
150 R=R-(CLA=6 AND R>0)+(CLA=7 AND R<36)
160 S=S-(CLA=14 AND S>0)+(CLA=15 AND S<21)
170 LOCATE R,S,G:POSITION R,S:PRINT CHR$(G);
180 T=T+(G=96)
190 POSITION 15,22:PRINT "Punteggio :";T
200 RETURN
210 A=X:B=Y
220 P=P-1
230 IF P=0 THEN GOSUB 500:PRINT "FINITO....":END
240 POSITION R,S:PRINT CHR$(160)
250 POSITION A,B:PRINT " ";CHR$(96);" "
260 X=ABS(X+INT(1-2*(RND(1)<0.5)))
270 Y=ABS(Y+INT(1-2*(RND(1)<0.5)))
280 IF X>36 THEN X=1
290 IF Y>21 THEN Y=1
300 RETURN
500 FOR I=0 TO 3
510 SOUND I,126-30*(I=1)-45*(I=2),10,4
520 FOR TE=1 TO 25:NEXT TE:NEXT I:SOUND 0,0,0,0:RETURN
```

Quattro a quattro

RIFERIMENTI:

Programmazione: difficile.
Categoria del gioco: riflessione (1 o 2 giocatori).
Difficoltà del gioco: elevata.

IL GIOCO:

Potrà essere fatto da uno o più giocatori. Nei due casi, lo scopo è di riuscire a cambiare le pedine X con le pedine O; se si gioca da soli, questo scambio dovrà essere realizzato col minor numero di mosse possibile.
Le pedine sono spostate alternativamente rispettando le regole di spostamento del cavallo negli scacchi; non ci sono prese e non si può passare.

IL PROGRAMMA:

Il ciclo I (60-100) disegna il tavolo di gioco e le posizioni di partenza.
La linea 120 permette di visualizzare O o X, secondo il giocatore che sta giocando.
La linea 150 permette la visualizzazione del cursore lampeggiante che indica su quale casella ci si trova.
Le linee 180-190 permettono lo spostamento del cursore.
La linea 200 guarda se è stato premuto il tasto che regola il gioco.
La linea 210 ristampa la figura che è sotto il cursore.
La linea 230 realizza il controllo di partenza e la 240 quello d'arrivo. La linea 250 verifica la validità della mossa. Il cambio del giocatore è realizzato dalla linea 270 e un "." è stampato al posto di partenza. Il sottoprogramma 330-340 è utilizzato per i diversi controlli.

PER GIOCARE:

Si lancia il programma con RUN. Appare il campo di gioco, così come il simbolo che permette di identificare il giocatore di turno. Il giocatore può scegliere la pedina che desidera spostare manovrando il joystick. Lo spostamento del cursore è evidenziato da un simbolo lampeggiante, battendo una prima volta sul bottone. La casella dalla quale si parte non ha più nessun simbolo. Manovrando il joystick il simbolo lampeggiante riappare e lo si posizionerà dove si vuole.
La mossa sarà rifiutata se la casella di partenza non è conforme o se la casella di arrivo è già occupata. Altrimenti la mossa è realizzata e la mano passa all'altro giocatore, che sposta il cursore lampeggiante fino a portarlo su una delle pedine e gioca allo stesso modo.

Il giocatore vincente è quello che riesce a portare per primo tutte le sue pedine nelle caselle inizialmente occupate dall'avversario. Si può interrompere la partita in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Si possono migliorare i controlli, prevedere un test di fine, utilizzare un campo di gioco differente; si potrà anche immaginare la scrittura di un programma che permetta all'ATARI di partecipare al gioco con delle buone possibilità di vincere.

```
10 REM QUATTRO A QUATTRO
20 GRAPHICS 2+16:SETCOLOR 4,2,14:SETCOLOR 1,1,2:SETCOLOR 2,2,14
40 H=1
50 POSITION 4,0:PRINT #6;"1 2 3 4 5"
60 FOR I=1 TO 5
70 POSITION 0,2*I:PRINT #6;STR$(I):POSITION 4,2*I:PRINT #6;". . . . ."
80 IF I<3 THEN POSITION 4,2*I:PRINT #6;"O O"
90 IF I>3 THEN POSITION 10,2*I:PRINT #6;"X X"
100 NEXT I
110 X=1:Y=1
120 POSITION 16,5:PRINT #6;CHR$(79+9*H);
125 X1=0
130 GOSUB 330
150 POSITION 2+2*X,2*Y:PRINT #6;" "
160 A=X:B=Y
170 R=STICK(0)
180 Y=Y-(R=14 AND Y>1)+(R=13 AND Y<5)
190 X=X-(R=11 AND X>1)+(R=7 AND X<5)
200 IF  NOT STRIG(0) THEN 230
210 POSITION 2+2*A,2*B:PRINT #6;CHR$(J)
220 GOTO 130
230 IF X1=0 AND J<>79+9*H THEN 210
240 IF X1=0 AND J=79+9*H THEN X1=X:Y1=Y:GOSUB 130
242 IF X1=0 THEN 210
245 IF J<>ASC(".") AND X1<>0 THEN GOTO 130
250 IF (ABS(X-X1)=1 AND ABS(Y-Y1)=2) OR (ABS(X-X1)=2 AND ABS(Y-Y1)=1) THEN 270
260 POSITION 2+2*X,2*Y:PRINT #6;".":X=X1:Y=Y1:GOTO 130
270 POSITION 2+2*X,2*Y:PRINT #6;CHR$(79+H*9):POSITION 2+2*X1,2*Y1:PRINT #6;".":H
= NOT H:FOR T=1 TO 10:NEXT T:GOTO 110
330 LOCATE 2+2*X,2*Y,J
340 RETURN
```

"Puzzle" di lettere

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: elevata.

IL GIOCO: Chi di noi non ha, un giorno o l'altro, avuto tra le mani uno di quei piccoli puzzles dove si cerca di spostare da una casella all'altra dei piccoli quadrati di plastica colorata, fino ad ottenere un risultato ben preciso? Al suo tempo il 14-15, di Sam Lloyd, ha avuto lo stesso successo del celebre Rubik's.
Il puzzle che è proposto qui utilizza 35 lettere, disposte a caso in un quadrato di 6*6 caselle, e queste lettere possono, riposizionate nel loro ordine iniziale, permettervi di ricostituire una frase.
Una casella lampeggiante vi permette di spostare le lettere ed un contatore visualizza in permanenza il numero di tentativi: riuscirete a trovare la frase?

IL PROGRAMMA: La frase da ricostituire (che potrà naturalmente essere modificata) è posizionata in una variabile R$ alla linea 60.
La mescola delle lettere e la stampa sono fatte dal ciclo I (70-160).
Le linee 190 realizza la visualizzazione del quadrato lampeggiante che è spostato dalle linee 230 e 240 utilizzando le quattro frecce.
Le linee 250 e 260 ristampano la lettera scelta nella posizione del quadrato colorato.
La linea 280 contabilizza il numero di tentativi, che è stampato grazie alla linea 180.

PER GIOCARE: Si lancia il programma con RUN.
La partita è più difficile se non si conosce la frase da ricostruire. Si possono spostare le lettere e farle passare nella posizione del quadrato colorato con l'aiuto dei 4 tasti di direzione (←, →, ↑, ↓).
Il numero di tentativi è visualizzato in permanenza, a destra del puzzle.
Si può fermare il gioco in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Lo stesso principio può essere applicato ad ogni sorta di puzzles (manipolazioni di numeri, di colori, di suoni eventualmente). Si possono anche prevedere varie frasi, con una scelta effettuata da un generatore casuale ecc.

```
10 REM PUZZLE DI LETTERE
20 DIM A$(35),R$(35)
30 GRAPHICS 0:POKE 752,1
35 SETCOLOR 1,3,3
40 FOR I=1 TO 8:POSITION 15+I,6:PRINT CHR$(160):POSITION 16,5+I:PRINT CHR$(160):
POSITION 15+I,13:PRINT CHR$(160)
50 POSITION 23,5+I:PRINT CHR$(160):NEXT I
60 A$="":R$="LAMARIANNAVAINCAMPAGNAINMONOPATTINO"
70 FOR I=1 TO 35
80 X=INT(RND(1)*35+1)
90 IF R$(X,X)="-" THEN 80
100 A$(LEN(A$)+1)=R$(X,X)
110 R$(X,X)="-"
120 F=ASC(A$(I,I))
130 X=I-6*INT(I/6):Y=INT(I/6)
140 POSITION 17+X,7+Y:PRINT CHR$(F)
160 NEXT I
170 X=0:Y=0:S=0:C=32
180 POSITION 0,22:PRINT "Numero di prove :";S
190 POSITION 17+X,7+Y:PRINT CHR$(32+FL)
200 CLA=PEEK(764):IF CLA=255 THEN FL=ABS(FL-128):GOTO 190
220 I=X:J=Y
230 X=X-(CLA=6 AND X>0)+(CLA=7 AND X<5)
240 Y=Y-(CLA=14 AND Y>0)+(CLA=15 AND Y<5)
250 LOCATE 17+X,7+Y,C
260 POSITION 17+I,7+J:PRINT CHR$(C)
270 IF I=X AND J=Y THEN SOUND 0,251,10,4:FOR TE=1 TO 5:NEXT TE:SOUND 0,0,0,0:GOT
O 290
280 S=S+1:FOR TE=1 TO 255 STEP 8:SOUND 0,I,10,4:NEXT TE:SOUND 0,0,0,0
290 POKE 764,255:GOTO 180
```

Bersagli

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: fortuna.
Difficoltà del gioco: bassa.

IL GIOCO: Consiste semplicemente nel colpire dei bersagli, cercando di segnare il maggior numero di punti; il numero di bersagli (da 1 a 6) sarà scelto all'inizio della partita ed il punteggio realizzato sarà visualizzato in permanenza. Se si tocca il bordo di un bersaglio, la partita si ferma. Si marcheranno tanti più punti quanto il numero colpito sarà grande.

IL PROGRAMMA: Il numero di bersagli è inserito alla linea 50. Il ciclo I (80-100) effettua la stampa dei bersagli.
La linea 120 visualizzerà il punteggio in permanenza.
La palla è visualizzata dalla linea 130.
Il ciclo I (150-200) realizza la stampa delle cifre contenute nei bersagli e permette lo spostamento della palla (linea 190).
La linea 220 guarda se si è sparato, e la linea 230 guarda se la palla è ancora sulla linea di partenza.
La linea 240 fa avanzare la palla di una casella verso destra, e la linea 250 verifica se ha oppure no raggiunto i bersagli; poi le linee 260 e 270 guardano ciò che hanno colpito.
La linea 280 incrementa il punteggio.
Le linee 290 e 300 si occupano della fine della partita.
Le note della musica sono scritte nella stringa VA$ con i valori dei suoni corrispondenti. Il sottoprogramma 500 sceglie alcune note.

PER GIOCARE: Lanciare il programma con RUN e battere il numero di bersagli desiderati seguito da RETURN. I bersagli appaiono e la partita comincia. La vostra pallina, raffigurata da una O è visualizzata in basso a sinistra; potete spostarla con le frecce alto e basso (tasti ← e →) e potete sparare quando volete battendo 0.
Il punteggio sarà modificato in funzione del vostro risultato, e una nuova pallina appare. È possibile fermarsi in ogni istante battendo BREAK. altrimenti il gioco si ferma dopo un certo numero di spari.

ESTENSIONI POSSIBILI: Si potrà senza difficoltà aumentare il numero di bersagli, modificare le regole di apparizione dei numeri nei bersagli per renderle meno casuali, fare in modo che la pallina stessa sia raffigurata da una cifra che dovrebbe coincidere all'arrivo con quella del bersaglio.

```
10 REM BERSAGLI
20 DIM MU$(100),VA$(45),C$(45)
25 VA$="DOREMIFASOLASIDMRM125108096091081072064063053":C$=VA$
30 GRAPHICS 0:POKE 752,1
40 PRINT "}"
50 PRINT "Numero di bersagli :";:INPUT N
55 IF N<0 OR N>6 THEN 40
60 P=10:J=30:S=0:B=1
70 PRINT "}"
80 FOR I=1 TO N
90 POSITION J,3*I-1:PRINT CHR$(18);CHR$(5):POSITION J,3*I:PRINT " |":POSITION J,
3*I+1:PRINT CHR$(18);CHR$(3);
100 NEXT I
110 K=0:A=2+INT(RND(1)*(3*N))
120 POSITION 0,20:PRINT "Punteggio :";S;" - Palla :";B
130 POSITION K,A:PRINT CHR$(20)
140 D=A
150 FOR I=1 TO N
160 U=INT(RND(1)*P)
170 POSITION J,3*I:PRINT CHR$(U+48)
180 CLA=PEEK(764)
190 A=A+(CLA=15 AND A<3*N+1)-(CLA=14 AND A>2)
195 POKE 764,255
200 NEXT I
210 POSITION K,D:PRINT " "
220 IF CLA=50 THEN B=B+1:GOTO 240
230 IF K=0 THEN 130
240 K=K+1
250 IF K<J THEN 130
260 LOCATE K,A,C:POSITION K,A:PRINT CHR$(C)
270 IF C<48 OR C>57 THEN POSITION 0,21:PRINT "Perso!....";:MU$="SOMIDO":GOSUB 50
0:END
280 S=S+C-48:MU$="DODODOREMIREDOMIREREDO":GOSUB 500
290 P=P-1
300 IF P=0 THEN MU$="SISISOSIRESOSOLADMSILASOL":GOSUB 500:END
310 GOTO 110
500 VA$=C$
505 FOR IM=1 TO LEN(MU$)-1 STEP 2
510 FOR JV=1 TO 9
520 IF VA$(2*JV-1,2*JV)<>MU$(IM,IM+1) THEN NEXT JV
530 NO=VAL(VA$(19+3*(JV-1),18+3*JV))
540 SOUND 0,NO,10,4:FOR TE=1 TO 10:NEXT TE
550 SOUND 0,0,0,0:FOR TE=1 TO 5:NEXT TE
560 NEXT IM
570 RETURN
```

Gocce

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO: È un vero programma di disegno animato che si può considerare, infatti, come un interludio; un recipiente è disegnato sullo schermo, visto di taglio; e contiene già degli ostacoli. Una goccia d'acqua, simbolizzata da un cerchio grigio, appare sopra il recipiente e cade riempiendolo man mano. La sua traiettoria, quando trova un ostacolo, è interamente determinata dal caso: va a destra o a sinistra, fino a che non non ci sia più alcuna possibilità. Allora, un'altra goccia cade, e così via, fino al traboccamento finale.

IL PROGRAMMA: I due cicli chiusi I (60-130) e J (70-120) disegnano il recipiente e gli ostacoli.
La linea 170 disegna la goccia e la linea 210 rinvia nel sottoprogramma (370-390) che permette di determinare se incontra o meno un ostacolo.
Se non l'incontra, si va in 290 e la goccia continua a cadere, altrimenti la goccia va a destra o a sinistra (linea 230) secondo il valore definito per K alla linea 150.
Si guarda di nuovo se c'è oppure no l'ostacolo, o se la goccia può continuare a cadere (linee 240, 250 o 260). Secondo il risultato del test, si utilizzerà 270, 290, 310 o 340.

PER GIOCARE: Il RUN fa partire il disegno del recipiente e degli ostacoli, la goccia appare, si mette a cadere e finisce per trovarsi un posto. Un'altra goccia appare e si ricomincia.
Il programma si ferma da solo quando il recipiente trabocca; possiamo fermarlo in qualsiasi momento battendo BREAK.

ESTENSIONI POSSIBILI: Possiamo, naturalmente, aggiungere il colore ed i suoni, ed anche immaginare un gioco nel quale un piccolo personaggio possa evitare le gocce o l'annegamento.

```
10 REM GOCCE
50 GRAPHICS 0:POKE 752,1:SETCOLOR 1,7,2:SETCOLOR 2,6,10
60 FOR I=12 TO 20
70 FOR J=1 TO 16
80 POSITION 0,I:PRINT CHR$(160):POSITION 17,I:PRINT CHR$(160)
90 POSITION J,20:PRINT CHR$(160)
100 IF I<>12 AND I<>20 THEN POSITION J,I:PRINT CHR$(160)
110 IF I<>20 AND RND(1)<0.7 THEN POSITION INT(RND(1)*15+1),I:PRINT "  "
120 NEXT J
130 NEXT I
140 Y=INT(RND(1)*7+6):X=6
150 K=2*(RND(1)<0.5)-1
160 I=0
170 POSITION Y,X:PRINT CHR$(20)
180 FOR J=0 TO 100:NEXT J
190 POSITION Y,X:PRINT " "
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN 310
260 IF I=0 THEN 340
270 POSITION Y,X:PRINT CHR$(20)
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 LOCATE V,U,C:POSITION V,U:PRINT CHR$(C):SOUND 0,31,10,4:FOR TE=1 TO 3:NEXT T
E:SOUND 0,0,0,0
380 J=(C<>32):IF U=23 THEN POSITION 18,15:PRINT "Finito.....":GOSUB 500:END
390 RETURN
500 FOR MU=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:F
OR TE=1 TO 5:NEXT TE:NEXT MU:RETURN
1000 DATA 63,47,40
```

Ostacoli

RIFERIMENTI: Programmazione: abbastanza difficile.
Categoria del gioco: gioco di riflessi (per un giocatore).
Difficoltà del gioco: media.

IL GIOCO: Dei pedoni si spostano sullo schermo incontro ad una piccola macchina che dispone della possibilità di spostarsi verso sinistra o destra per evitarli il più a lungo possibile.
La distanza percorsa sarà visualizzata in permanenza ed il gioco termina quando la macchina investe un pedone o il bordo della strada.

IL PROGRAMMA: La linea 80 fa avanzare i pedoni verso la macchina del giocatore che è stampata grazie alla linea 70.
Le linee 80, 90 e 100, disegnano la strada (le linee 90 e 100, anche se identiche, sono tutte e due indispensabili, poiché il valore di RND non sarà, in teoria, lo stesso sulle due linee).
La linea 120 visualizza il punteggio.
La linea 140 permetterà di spostare il veicolo.
Le linee 150 e 160 permettono di controllare se c'è o no un ostacolo e fermano il programma se è necessario.
Il punteggio è incrementato dalla linea 170.

PER GIOCARE: Si lancia il programma battendo RUN.
La strada e i pedoni appaiono nella parte inferiore dello schermo e salgono verso la macchina. La macchina può spostarsi con l'aiuto delle frecce di spostamento orizzontale (tasti ← e →).
Lo scontro con un ostacolo interrompe lo svolgimento del programma e provoca la visualizzazione del punteggio.
Si può, però, uscire dal programma in ogni momento utilizzando il tasto BREAK.

ESTENSIONI POSSIBILI: Sarà abbastanza facile migliorare il disegno della macchina, della strada, e degli ostacoli. Potremo migliorare il suono e il colore, ammettere un certo numero di incidenti, registrare il miglior punteggio (per vari giocatori), eventualmente disegnare varie macchine, ecc.

```
10 REM OSTACOLI
50 GRAPHICS 0:POKE 752,1:SETCOLOR 1,3,10:SETCOLOR 2,0,4:SETCOLOR 4,0,4
60 S=0:Y=5
70 POSITION Y,0:PRINT "Y";
80 POSITION 0,23:PRINT CHR$(160);"        ";CHR$(160);:REM 8 BLANCS
90 IF RND(1)<0.7 THEN POSITION INT(RND(1)*8+1),23:PRINT CHR$(123);
100 IF RND(1)<0.8 THEN POSITION INT(RND(1)*8+1),23:PRINT CHR$(123)
110 POSITION 15,5:PRINT "               ":REM 15 BLANCS
120 POSITION 15,6:PRINT "Puntessio :";S
130 CLA=PEEK(764)
140 Y=Y-(CLA=6 AND Y>0)+(CLA=7 AND Y<9)
150 LOCATE Y,1,C:POSITION Y,1:PRINT CHR$(C)
160 IF C=160 OR C=123 THEN GOSUB 500:END
170 S=S+1
180 POKE 764,255:GOTO 70
500 FOR MU=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:F
OR TE=1 TO 5:NEXT TE:NEXT MU:RETURN
1000 DATA 63,53,47
```

Catture nello spazio

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessi (un solo giocatore).
Difficoltà del gioco: media.

IL GIOCO: La vostra astronave naviga in un pericoloso campo di meteoriti e la vostra missione è di impadronirvi della maggiore quantità possibile di minerale di uranio, un metallo indispensabile che si trova solo in questo luogo pericoloso. Dovete riuscire a manovrare per raccogliere l'uranio, senza farvi urtare da una meteorite.

IL PROGRAMMA: La linea 70 permette di fare apparire le meteoriti.
La linea 90 fa apparire il minerale di uranio e assicura lo spostamento delle meteoriti sullo schermo.
Le linee 110-130 controllano lo svolgimento del gioco (abbiamo urtato una meteorite? o abbiamo preso un blocco di uranio?).
Lo scontro con una meteorite provoca la fine della partita.
La vostra astronave è stampata dalle linee 140 e 150 ed il suo spostamento è controllato dalla linea 170.

PER GIOCARE: Si lancia il programma con RUN e le meteoriti cominciano a salire sullo schermo, così come i blocchi di uranio.
I tasti di spostamento orizzontale (← e →) saranno utilizzati per spostare la vostra astronave e recuperare l'uranio, ma il minimo urto con una meteorite provocherà la fine della partita e la visualizzazione della quantità di uranio catturata. Si potrà anche fermare in ogni momento battendo BREAK.

ESTENSIONI POSSIBILI: Si può rendere il gioco sempre più difficile aumentando il numero di meteoriti, o ancora, dando un limite di tempo alla partita.
Si può anche prevedere una visualizzazione permanente del punteggio, introdurre una seconda astronave manovrata da una potenza straniera che cercherà anch'essa di impadronirsi dell'uranio...

```
10 REM CATTURE NELLO SPAZIO
20 REM
30 GRAPHICS 0:POKE 752,1
40 SETCOLOR 1,2,10:SETCOLOR 2,7,2:SETCOLOR 4,7,2
50 X=15:Z=0
60 S=INT(RND(1)*39)
70 POSITION S,23:PRINT "#";
80 S=INT(RND(1)*39)
90 IF RND(1)<0.7 THEN POSITION S,23:PRINT "*";:PRINT :PRINT
100 FOR I=1 TO 10:NEXT I
110 LOCATE X,13,C:POSITION X,12:PRINT CHR$(C);
120 IF C=42 THEN POSITION 10,23:PRINT "Punteggio ottenuto :";Z:RESTORE 1000:GOSU
B 500:END
130 IF C=35 THEN Z=Z+1:RESTORE 2000:GOSUB 500
140 POSITION X-1,10:PRINT "   ";
150 POSITION X-1,12:PRINT " ";CHR$(0);" ";
160 CLA=PEEK(764)
170 X=X-(CLA=6 AND X>1)+(CLA=7 AND X<37)
180 POKE 764,255:GOTO 60
500 FOR MU=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:F
OR TE=1 TO 5:NEXT TE:NEXT MU:RETURN
1000 DATA 63,47,40
2000 DATA 126,108,96
```

Anomalia

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Si basa essenzialmente su una buona capacità di osservazione. Due griglie di 25 caselle ciascuna sono disegnate sullo schermo. Dei simboli identici sono visualizzati in posti differenti delle due griglie, ma uno dei simboli è stato modificato sulla griglia di destra. Il gioco consiste nel trovare quale, inserendo le sue coordinate nell'ordine linea-colonna.

IL PROGRAMMA: Il ciclo I (50-80) stampa le cifre di riferimento delle griglie.
Il ciclo principale I (100-160) stampa i simboli delle due griglie, in un ordine differente.
La linea 110 sceglie il carattere da stampare.
Il sottoprogramma (1000-1040) si occupa di fare la stampa in una casella libera (lo si richiama due volte per le due griglie).
Le linee 170 e 180 cambiano, in maniera casuale, uno dei caratteri.
La linea 170 toglie il colore dello schermo scelto alla linea 45.
Le coordinate del punto tentato sono inserite alle linee 190 e 200, ed il controllo è effettuato dalle linee 210 e 220.
Le linee 230 e 240 si occupano della visualizzazione del risultato.

PER GIOCARE: Dopo aver lanciato il programma con l'istruzione RUN, si ottiene la visualizzazione di due griglie. Si dovrà allora cercare, per paragoni, qual è il carattere che è stato modificato (sulla griglia di destra).
Si inserisce allora il suo numero di linea, seguito da RETURN, ed il suo numero di colonna, seguito da RETURN.
Il programma risponderà allora BRAVO o la lettera da trovare era . . .

ESTENSIONI POSSIBILI: Sarà possibile utilizzare dei simboli grafici di differenti colori, il che potrà semplificare la ricerca. Si può anche immaginare di utilizzare dei disegni più sofisticati, migliorare il controllo dei risultati, ammettere la risposta esatta sulla griglia di destra o di sinistra, introdurre un controllo di tempo, aggiungere il suono ecc.

```
10 REM ANOMALIA
30 DIM I$(1)
40 GRAPHICS 1:T=5
45 SETCOLOR 0,0,0
50 FOR I=1 TO 5
55 I$=CHR$(I+48+128)
60 POSITION I+1,0:PRINT #6;I$:POSITION 0,2*I:PRINT #6;I$
70 POSITION I+10,0:PRINT #6;I$:POSITION 9,2*I:PRINT #6;I$
80 NEXT I
100 FOR I=1 TO 22
110 D=INT(26*RND(1)+1)
120 L=2:S=T
130 GOSUB 1000
140 L=11
150 GOSUB 1000
160 NEXT I
170 S=0:SETCOLOR 0,10,8
180 GOSUB 1000
190 PRINT "Linea.....:";:INPUT I
200 PRINT "Colonna...:";:INPUT T
210 U=2*I:V=T+10
220 LOCATE V,U,C:POSITION V,U:PRINT CHR$(C)
230 IF D+64=C THEN PRINT "BRAVO! ";:FOR I=1 TO 3:SOUND 0,63*(I=1)+47*(I=2)+40*(I
=3),10,4:FOR T=1 TO 5:NEXT T:NEXT I
240 PRINT "La lettera da trovare era ";CHR$(D+64):END
1000 U=2*INT(RND(1)*T+1):V=INT(RND(1)*S+L)
1010 LOCATE V,U,C:POSITION V,U:PRINT #6;CHR$(C)
1020 IF C<>32 AND S<>0 THEN 1000
1030 POSITION V,U:PRINT #6;CHR$(D+64)
1040 RETURN
```

Tanti in uno

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: due giocatori.
Difficoltà del gioco: secondo l'utilizzo.

IL GIOCO: Molti giochi sono in realtà possibili con questo programma che si comporterà come una matita elettronica, che vi permetterà di scrivere dei simboli O o X , in un qualunque posto del vostro foglio-schermo. Potremo, per esempio, utilizzarlo per un gioco da ragazzi (il primo a realizzare un allineamento di 5 pedine identiche ha vinto) o per un GO (si realizzano gli accerchiamenti delle pedine avversarie) o un altro gioco, secondo la vostra immaginazione.

IL PROGRAMMA: Il campo di gioco è delimitato da un quadro posizionato alla linea 30 e 35.
Il cambio del giocatore è effettuato dalla linea 50.
Il giocatore di turno vede il simbolo che ha scelto (O o X) spostarsi al limite del quadrato (linea 70).
Le linee 100 e 110 permettono ai due giocatori di spostare il simbolo sul terreno.
Le linee 130 e 140 controllano se lo spostamento è possibile (casella libera e non oltrepasso dei limiti).
La linea 190 permette di scrivere in una casella libera.

PER GIOCARE: RUN provoca l'apparizione dei limiti del campo e del numero del giocatore, così come i simboli che si dovranno spostare sullo schermo con l'aiuto dei 4 tasti di direzione (← → ↑ ↓). Non è possibile andare in una casella già occupata e bisognerà battere il tasto 0 per iscrivere il proprio simbolo in una casella disponibile.
Fatto questo il simbolo del secondo giocatore appare nella colonna di sinistra, circa alla metà, ed il giocatore procederà nella stessa maniera del suo avversario.
Per uscire dal programma, basterà battere BREAK.

ESTENSIONI POSSIBILI: Si potrà utilizzare l'intero schermo come terreno di gioco e realizzare dei controlli in funzione dei differenti giochi che si desiderano fare: controllo di allineamento di 5 pedine per il gioco di cui parlavamo prima, controllo del numero di pedine catturate per il GO.

```
10 REM TANTI IN UNO
15 OPEN #1,4,0,"K:"
20 GRAPHICS 0:POKE 752,1
25 SETCOLOR 1,0,0:SETCOLOR 2,1,10:SETCOLOR 4,1,10
30 POSITION 0,0:PRINT CHR$(17):POSITION 0,23:PRINT CHR$(26);
35 FOR I=1 TO 22:POSITION I,0:PRINT CHR$(18):POSITION 0,I:PRINT "|":POSITION I,2
3:PRINT CHR$(18);:NEXT I
40 J=0
50 J=ABS(J-1)
60 X=12:Y=1
70 POSITION Y,X:PRINT CHR$(79+9*(J=1))
80 GET #1,CLA
90 A=X:B=Y
100 X=X+(CLA=61 AND X<22)-(CLA=45 AND X>1)
110 Y=Y+(CLA=42 AND Y<22)-(CLA=43 AND Y>1)
120 POSITION B,A:PRINT " "
130 LOCATE Y,X,C:POSITION Y,X:PRINT CHR$(C)
140 IF C<>79 AND C<>88 THEN 160
150 X=A:Y=B
160 FOR I=1 TO 50:CLA=PEEK(764)
170 IF CLA=255 THEN NEXT I
180 POKE 764,255:IF CLA<>50 OR Y=0 THEN 70
190 POSITION Y,X:PRINT CHR$(79+9*(J=1));
200 SOUND 0,63*(J=0)+40*(J=1),10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0
210 GOTO 50
```

Percorso del cavaliere

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: elevata.

IL GIOCO: Si presenta, qui, come un gioco totalmente d'azzardo, ma proviene, in realtà, da un problema assai difficile da risolvere: come spostare un cavallo su una scacchiera in modo che passi una volta, e una sola, su ognuna delle 64 caselle della scacchiera.
Il cavallo si sposta secondo le regole comunemente ammesse negli scacchi e la sua posizione di partenza è definita dal giocatore; i suoi spostamenti, invece sono effettuati a caso. Riuscirà a percorrere le 64 caselle?

IL PROGRAMMA: La linea 60 definisce i diversi spostamenti possibili, in due variabili alfanumeriche A$ e B$; il ciclo chiuso I (70-100) disegna il campo di gioco.
Si inseriscono le coordinate della casella di partenza (linea poi colonna) alle linee 100 e 120.
La linea 140 stampa il numero della mossa.
La linea 150 permette la pesca a caso di uno spostamento del cavallo.
Il ciclo ZE (160-210) cerca se lo spostamento è possibile; se non è il caso, si cercano tutte le altre (sempre nello stesso ordine).
Le linee 170 e 180 effettuano lo spostamento.
La linea 190 controlla che non si esca dalla scacchiera.
La linea 200 invia alla linea 230 se il colpo è possibile.
La linea 220 gestisce la fine della partita (cavallo bloccato) e visualizza il punteggio.
Le linee 230, 240 e 250 realizzano la mossa e incrementano il punteggio.
La linea 500 evita un messaggio d'errore e l'interruzione del programma quando leggendo la linea 2000 gli manca un dato.

PER GIOCARE: Si lancia il programma con l'ordine RUN. Appare il campo di gioco; si sceglie la posizione di partenza battendo prima un numero di linea (tra 1 e 8), poi il suo numero di colonna, seguito da RETURN. Il gioco comincia immediatamente e si svolge in modo interamente automatico fino a che il cavallo sia bloccato.
Si può, naturalmente, fermarsi in ogni momento con BREAK.

ESTENSIONI POSSIBILI: Si possono immaginare numerose varianti su questo tema, cominciando col far variare le dimensioni della scacchiera, si può anche migliorare considerevolmente la grafica. Ma, la cosa più interessante, è di riuscire a definire un algoritmo che dia la o le soluzioni del problema iniziale...

```
10 REM PERCORSO DEL CAVALIERE
15 GRAPHICS 0:POKE 752,1
20 DIM A$(16),B$(16),C$(16)
30 SETCOLOR 1,11,2:SETCOLOR 2,11,10
50 S=1:T=0
60 A$="56653223":B$="653223":B$(LEN(B$)+1)=A$:B$(LEN(B$)+1)="65":A$(LEN(A$)+1)=A
$
65 C$=B$
70 FOR I=1 TO 8
80 POSITION 3*I+1,0:PRINT I
90 POSITION 1,2*I:PRINT I;"  .  .  .  .  .  .  .  ."
100 NEXT I
110 POSITION 2,19:PRINT "Linea di partenza   :";:INPUT I
120 PRINT :PRINT "Colonna di pertenze :";:INPUT J
130 IF S>9 THEN S=0:T=T+1
140 POSITION 3*J+1,2*I:PRINT STR$(S):IF T<>0 THEN POSITION 3*J,2*I:PRINT STR$(T)
150 Z=INT(RND(1)*8+1)
160 FOR ZE=Z TO Z+8
170 A=I+VAL(A$(ZE,ZE))-4
180 B=J+VAL(B$(ZE,ZE))-4
190 IF A<1 OR B<1 OR A>8 OR B>8 THEN 210
200 LOCATE 3*B+1,2*A,C:POSITION 3*B+1,2*A:PRINT CHR$(C):B$=C$:IF C=46 THEN 230
210 NEXT ZE
220 POSITION 25,18:PRINT "Totale :";T*10+S:RESTORE 1000:GOSUB 500:END
230 I=A:J=B
240 S=S+1:RESTORE 2000:GOSUB 500
250 GOTO 130
500 TRAP 520
510 FOR MU=1 TO 3:READ VA:SOUND 0,VA,10,4:FOR TE=1 TO 10:NEXT TE:SOUND 0,0,0,0:F
OR TE=1 TO 5:NEXT TE:NEXT MU
520 RETURN
1000 DATA 126,196,81
2000 DATA 63,40
```

Muro di mattoni

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessi (per 1 giocatore).
Difficoltà del gioco: elevata.

IL GIOCO: Se c'è un gioco che si può presentare come un grande classico dei videogiochi per microcalcolatore, è proprio il muro di mattoni (in inglese BREAKOUT).
È offerto, qui, in una versione semplificata, ma è ugualmente interessante: visualizzazione del punteggio a fine partita, rimbalzi casuali ecc.
La regola è molto elementare: disponete di una racchetta che può essere spostata orizzontalmente per rinviare una pallina (materializzata da un punto) che demolirà i mattoni disegnati nella parte alta dello schermo.
Il vostro problema è di demolirne il maggior numero possibile, senza mancare la pallina, che provocherebbe la fine della partita e la visualizzazione del punteggio.

IL PROGRAMMA: Il ciclo I (50-80) disegna il campo di gioco ed i mattoni sono messi in ordine dai cicli chiusi I (90-130) e J (100-120).
La linea 150 disegna la racchetta la cui posizione è controllata dalla linea 170.
Le linee 220-250 guardano se si è colpito il muro, i mattoni o la racchetta.
I rimbalzi sono gestiti di conseguenza in maniera eventualmente casuale dalle linee 260-290.
La linea 190 visualizza il punteggio in permanenza.

PER GIOCARE: Il programma sarà lanciato dall'istruzione RUN, il che provoca l'apparizione del campo di gioco, poi, molto rapidamente, l'invio della pallina. Il giocatore dovrà allora cercare di spostare la sua racchetta con l'aiuto dei tasti di spostamento orizzontale (← e →).
Se, in un momento qualunque, la pallina non tocca la racchetta, il punteggio apparirà battendo due volte un tasto e la partita riprende battendo nuovamente un tasto. Si può anche terminare in ogni istante con BREAK.

ESTENSIONI POSSIBILI: Si può immaginare di ingrandire il campo di gioco, aumentare il numero di mattoni, prevedere varie palline, introdurre degli effetti sonori.
Si può anche, ispirandosi ai giochi dei bar, introdurre dei piccoli operai che verranno a ricostruire il vostro muro di mattoni, man mano che voi lo demolite.

```
10 REM MURO DI MATTONI
20 DIM A$(1)
40 GRAPHICS 0:A$=CHR$(160):POKE 752,1
45 SETCOLOR 1,2,2:SETCOLOR 2,11,10:SETCOLOR 4,11,10
50 FOR I=1 TO 18
60 POSITION 9+I,0:PRINT A$
70 POSITION 10,I:PRINT A$:POSITION 27,I:PRINT A$
80 NEXT I
90 FOR I=1 TO 7
100 FOR J=11 TO 26
110 POSITION J,I:PRINT "#"
120 NEXT J
130 NEXT I
140 H=-1:K=0.5:J=18:I=17:Y=J
150 POSITION Y-2,19:PRINT " --- "
160 CLA=PEEK(764)
170 Y=Y+(CLA=7 AND Y<28)-(CLA=6 AND Y>8)
180 IF T=1 THEN POSITION J,I:PRINT " "
190 POSITION 0,20:PRINT "Punteggio ";S
200 IF I>21 THEN FOR MU=0 TO 5:SOUND 0,63,10,4*(MU/2=INT(MU/2)):FOR TE=1 TO 5:NE
XT TE:NEXT MU:END
210 I=I+H:J=J+K
220 LOCATE J,I,C:POSITION J,I:PRINT CHR$(C);
230 IF C=32 THEN POSITION J,I:PRINT CHR$(20):T=1:GOTO 150
240 IF C=160 THEN 280
250 IF C=35 THEN POSITION J,I:PRINT CHR$(20):T=1:GOTO 300
260 H=-H
270 K=RND(1)-RND(1)
280 K=-K:T=0
290 GOTO 150
300 S=S+9
310 K=-(K-(1-2*RND(1)<0.5)*RND(1)/3)
320 H=-H
330 GOTO 150
```

Othello

RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: elevata.

IL GIOCO: Diventato un grande classico dei giochi su microcalcolatore, il gioco Othello o Reversi si avvantaggia dell'essere gestito su calcolatore con visualizzazione; anche se, come qui, è praticato da due giocatori: in effetti, il gioco consiste essenzialmente nell'accerchiare le pedine dell'avversario, e questo può rivelarsi alla lunga molto fastidioso, soprattutto se siete un giocatore assiduo.
Il principio è che il giocatore con le X deve, per poter giocare, riuscire a chiudere una o più pedine dell'avversario, in una o più direzioni; se ci riesce, le O saranno convertite in X (la conversione potrà essere fatta orizzontalmente, verticalmente, o seguendo le diagonali).
Il giocatore che ha le O procederà nella stessa maniera, finché la scacchiera sarà riempita o finché nessuno possa più giocare (se un giocatore non può aggirare le pedine, deve cedere il turno).

IL PROGRAMMA: I cicli chiusi (50-120) e Q (70-110) disegnano il campo di gioco e la linea 140 visualizza una X o una O, secondo il giocatore che inizia.
Il ciclo P (200-320) controlla la validità del colpo (la casella è libera: linee 300 e 310), utilizzando una parte del programma che si trova alle linee 360-420 (avete preso uno o più nemici in linea 360 ed eventuale ritorno autorizzato alla linea 380; si fa una pulizia in tutte le direzioni, se l'inquadramento si è rivelato possibile).
La linea 200 definisce le 8 direzioni di esplorazione.
Le linee 240 e 250 calcolano gli spostamenti da effettuare per esplorare queste 8 direzioni.
La linea 230 realizza gli spostamenti.
La linea 340 segnala il cambiamento del giocatore, se c'è stata conversione (controllo alla linea 330).

PER GIOCARE: Dopo RUN, ogni giocatore inserirà a turno le coordinate (linea, RETURN, colonna, RETURN) dellá casella dove si vuole giocare; se il colpo è valido, le pedine sono girate e si cambia il giocatore; se il colpo non è possibile si inserisce 0 come prima coordinata per il cambio del giocatore.

ESTENSIONI POSSIBILI: Prima di tutto, naturalmente, migliorare la visualizzazione con dei colori e fare una griglia più spaziosa.
Si potrà anche immaginare un gioco macchina per l'ATARI.
Dopo aver lanciato il programma con RUN, conviene inserire le dimensioni della griglia (4, RETURN o 6, 8 ...).

```
10 REM OTHELLO
15 DIM C$(1):GRAPHICS 0
20 PRINT "Dimensione della griglia :";:INPUT A
30 IF A/2<>INT(A/2) THEN 20
40 PRINT "}":POKE 752,1
50 FOR P=1 TO A
60 POSITION 2*P,0:PRINT STR$(P)
70 FOR Q=1 TO A
80 POSITION 0,2*P:PRINT STR$(P):POSITION 2*Q,2*P:PRINT "."
90 IF Q=A/2+1 AND P=A/2 THEN POSITION 2*Q-2,2*P:PRINT "O X"
100 IF Q=A/2+1 AND P=A/2+1 THEN POSITION 2*Q-2,2*P:PRINT "X O"
110 NEXT Q
120 NEXT P
130 K=1
140 POSITION 24,5:PRINT "Giocatore ";:C$="O":IF K=1 THEN C$="X"
145 PRINT C$:FOR I=1 TO 5:SOUND 0,63,10,4:NEXT I:SOUND 0,0,0,0
150 POSITION 31,8:PRINT "      ":POSITION 31,10:PRINT "      "
160 POSITION 23,8:PRINT "Linea  :";:INPUT U
170 IF U=0 THEN 340
180 POSITION 23,10:PRINT "Colonna:";:INPUT V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U:R=2*V
240 F=(P<3 OR P>7)-(P>3 AND P<7)
250 G=(P>=6)-(P>1 AND P<5)
260 Q=46+42*(S=1)+33*(S=2)
270 IF S=0 THEN 290
280 POSITION R,T:PRINT CHR$(Q)
290 T=T+2*F:R=R+2*G
300 LOCATE R,T,C:POSITION R,T:PRINT CHR$(C)
310 IF C=79 OR C=88 THEN 360
320 NEXT P
330 IF H<>0 THEN 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=79 AND K=1) OR (C=88 AND K=0) THEN W=1
370 IF S<>0 THEN POSITION R,T:PRINT CHR$(Q)
380 IF W=1 AND ((C=79 AND K=1) OR (C=88 AND K=0)) THEN 290
390 IF W=0 OR S<>0 THEN 320
400 S=1+(K=0)
410 H=H*(S=0)
420 GOTO 220
```

"Puzzle" dispettoso

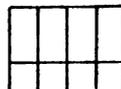
RIFERIMENTI: Programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: S'ispira al Puzzle 14-15 di Sam Lloyd ma qui le cifre sono sostituite da lettere.

IL PROGRAMMA: Il sottoprogramma 1000 disegna il terreno di gioco
La linea 130 inizializza la matrice A che contiene il valore della lettera visualizzata sulla griglia.
Il sottoprogramma 2000 visualizza questa lettera.
La linea 170 stampa 2 linee di istruzione; la linea 200 assicura lo spostamento del buco e TRAP permette il passaggio agli indici della matrice A per spostare il buco nei limiti della tabella.
Le linee 230-240 cambiano le lettere e il buco.

PER GIOCARE: Si lancia il programma con RUN. Si sposta il buco con l'aiuto delle frecce. Attenzione, il buco si sposta in senso inverso rispetto alle frecce: come nella realtà se si deve spostare una lettera verso sinistra (per esempio) si sposta il buco verso destra.
Per interrompere battere BREAK.

ESTENSIONI POSSIBILI: Migliorare i suoni, utilizzare il joystick per spostare il buco.

```
5 REM PUZZLE DISPETTOSO
10 CC=3:LC=3:GRILLE=1000:CASE=2000:DIM D$(9):DIM A(3,3)
20 GRAPHICS 2:POKE 756,226:SETCOLOR 0,3,8:SETCOLOR 4,3,8:GOSUB GRILLE
110 FOR C=0 TO 3
120 FOR L=0 TO 3
130 A(C,L)=225+4*C+L:IF C=3 AND L=3 THEN A(C,L)=32
140 X=C:Y=L:GOSUB CASE
150 NEXT L
160 NEXT C
170 POKE 752,1:PRINT :PRINT "      PER SPOSTARE LE LETTERE"
180 PRINT "   BATTERE I 4 TASTI CURSORE";
200 K=PEEK(764):POKE 764,255:TRAP 200:IF K<>6 AND K<>7 AND K<>14 AND K<>15 THEN
200
210 SOUND 0,45,10,10:FOR T=1 TO 10:NEXT T:SOUND 0,0,0,0
220 CN=CC+(K=6)-(K=7):LN=LC+(K=14)-(K=15)
230 A(CC,LC)=A(CN,LN):X=CC:Y=LC:GOSUB CASE
240 CC=CN:LC=LN:A(CC,LC)=32:X=CC:Y=LC:GOSUB CASE
250 GOTO 200
1000 HAUT=10000:BAS=10030:INTER=10020:LIGNE=10010
1010 FOR I=0 TO 8
1020 X=HAUT*(I=0)+LIGNE*(I/2<>INT(I/2))+INTER*(I/2=INT(I/2) AND I>0 AND I<8)+BAS
*(I=8)
1030 RESTORE X:READ D$:POKE 85,6:PRINT #6;D$
1040 NEXT I
1050 RETURN
2000 POSITION 7+2*X,2*Y+1:PRINT #6;CHR$(A(X,Y)):RETURN
10000 DATA qrwrwrwre:.battere simultaneamente <CONTROL>
10010 DATA | | | | |
10020 DATA arsrsrsrd:.vedere 10000
10030 DATA zrxrxrxrc:.vedere 10000
```

Super scorrimento

RIFERIMENTI: Programmazione: di media difficoltà.
Categoria del gioco: animazione.
Difficoltà del gioco: nessuna.

IL GIOCO: Si deve ricostruire uno schermo, con le lettere pari che scendono e le lettere dispari che salgono, prima di farle nascondere.

IL PROGRAMMA: La stringa A$ contiene il messaggio da far scorrere.
T$ rappresenta la stringa visualizzata sullo schermo.
TRAP in 1040 evita la visualizzazione fuori dallo schermo.

PER GIOCARE: Si lancia il programma con RUN e lo schermo si anima.
I colori cambiano automaticamente.
Si interrompe con BREAK.

ESTENSIONI POSSIBILI: Si può cambiare la frase in DATA, immaginare spostamenti più complessi (giro dello schermo, spirale ecc...). Sarà divertente guardare l'influenza del testo battuto in maiuscole, minuscole con colore, così come mettere o meno i caratteri inversi.

```
5 REM SUPER SCORRIMENTO
10 DIM A$(99):DIM TI$(20):DIM C$(10)
20 DUREE=40:ATTENTE=3000
30 GRAPHICS 2+16
40 READ A$
100 REM Visualizzazione di A$(1,20)sulla linea LARRIVEE, le lettere dispari sala
ono,quelle pari scendono
110 TI$=A$(1,20)
120 LDEPART=0:LARRIVEE=5:FOR C=0 TO 19 STEP 2:GOSUB 1000:NEXT C
130 LDEPART=11:LARRIVEE=5:FOR C=1 TO 19 STEP 2:GOSUB 1000:NEXT C
200 REM Scorrimento di A$ verso sinistra
210 I=21:GOSUB 2000
220 DUREE=1000:GOSUB ATTENTE:DUREE=40:PRINT #6;"}"
300 REM Scelta di 5 nuovi colori diversi tra gli 8 di base(0..7)rappresentati da
 C$
305 REM Ogni colore scelto in C e' annullato in c$
310 C$="01234567"
320 FOR J=1 TO 5
330 C=INT(RND(1)*8):IF C$(C+1,C+1)=" " THEN 330
340 C$(C+1,C+1)=" ":POKE 707+J,2*C
350 NEXT J
400 GOTO 100
1000 REM Scorrimento del C+1esimo carattere di TI$ tra la linea LDEPART e la lin
ea LARRIVEE sulla colonna C
1010 S=SGN(LARRIVEE-LDEPART)
1020 FOR L=LDEPART TO LARRIVEE STEP S
1030 POSITION C,L:PRINT #6;TI$(C+1,C+1);
1040 TRAP 1050:POSITION C,L-S:PRINT #6;" "
1050 TRAP 45000:NEXT L
1060 RETURN
2000 REM Visualizzazione di TI$ sulla linea LARRIVEE con scorrimento verso`sinis
tra pr riempire A$
2010 FOR J=I TO LEN(A$)
2020 TI$=TI$(2,20):TI$(20,20)=A$(J)
2030 POSITION 0,LARRIVEE:PRINT #6;TI$
2040 GOSUB ATTENTE
2050 NEXT J
2060 RETURN
3000 REM Sottoprogramma d'attesa
3010 FOR T=1 TO DUREE:NEXT T:RETURN
10000 DATA BUONGIORNO MIO CARO GIACOMO COME stai? Qui tutto LISCIO come l'olio.
20000 REM Il testo da fa scorrere e' scritto sottoforma di DATA
```

Insegnare e divertire è il doppio obbiettivo di questo libro. I 102 programmi di giochi qui contenuti vi invitano alla scoperta del'ATARI (600 XL, 800 XL e 130 XE). La classificazione per livelli di difficoltà crescente vi permette di acquisire a poco a poco nuove conoscenze e una maggiore padronanza del Basic.
Per ogni livello la descrizione del giochi proposti è seguita da un programma commentato in modo esauriente e da un esempio di utilizzazione.
L'ultimo livello costituisce un'illustrazione chiara dell'impiego del Basic esteso.

★★

L 22.000

I.S.B.N. 88.7688.**104**.2

9 788876 881046

102 PROGRAMMI PER ATARI
JACQUES DECONCHAT
E.P.S.I.